

IDENTIFICACIÓN Y CONTROL DE UNA TURBINA EÓLICA A PEQUEÑA ESCALA MEDIANTE EL CONJUNTO DE FILTROS DE KALMAN (EnKF)

Omar Ernesto López Santacruz

Trabajo de grado presentado como requisito
parcial para optar al título de
Ingeniero Electricista

Director
Eduardo Giraldo Suarez

Pereira, junio de 2015
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
Programa de Ingeniería Eléctrica.
Grupo de investigación en automática



Nota de Aceptación

Eduardo Giraldo Suarez. Director

Jurado

Pereira, junio de 2015

IDENTIFICACIÓN Y CONTROL DE UNA TURBINA EÓLICA A PEQUEÑA ESCALA
MEDIANTE EL CONJUNTO DE FILTROS DE KALMAN (EnKF)
©Omar Ernesto López Santacruz

Pereira, junio de 2015
Programa de Ingeniería Eléctrica.
Universidad Tecnológica de Pereira
La Julita. Pereira(Colombia)
TEL: (+57)(6)3137122
www.utp.edu.co
Versión web disponible en: *<http://recursosbiblioteca.utp.edu.co/tesis/index.html>*

Dedicatoria

A mi madre Ana Rocío Santacruz Arciniegas, a quien debo lo que soy y lo que tengo, a lo largo de mi vida y de mi carrera he contado con su afecto y apoyo incondicional, sus palabras y su cariño me han hecho no desfallecer ni darme por vencido en todas las metas que me he propuesto.

Agradecimientos

Quiero dar las gracias a mi director de proyecto de grado Eduardo Giraldo Suarez por su gran apoyo y por haberme tenido en cuenta en su grupo de investigación, por admiración hacia quienes trabajan en el área de control, tome la decisión de realizar mi proyecto en esta temática tan apasionante, gracias a mis compañeros Mario Inguilan, Daniel Duran, Fredy Arboleda, Fredy Cuellar, Yeison Rotavista, Lorena Castro, Jonathan Peña, Javier Pinzón, Daniel ipus, Yareth Perdomo, Andres Marín Y Felipe Ayala, por brindarme su amistad a lo largo de mi carrera, gracias a todos mis profesores por su dedicación y sus enseñanzas, gracias a mi familia, finalmente quiero dar las gracias a mi novia Cristina Delgado y a mi primo Sebastian Muñoz por su compañía, apoyo y cariño.

Resumen

Empleando el filtro de Kalman estándar se aborda el problema de la estimación de estados, para posteriormente abordar el problema de la estimación de parámetros o identificación de sistemas, finalmente se desarrolla el filtro de Kalman dual encargado de realizar las dos estimaciones de manera simultánea, los algoritmos desarrollados en esta instancia funcionan de manera óptima, sin embargo su funcionamiento se encuentra restringido a sistemas lineales.

En segundo lugar partiendo del filtro de Kalman extendido se busca abordar las problemáticas mencionadas para sistemas no lineales, por ende a partir del operador jacobiano se linealiza el modelo del sistema con el objetivo de emplear las ecuaciones del filtro de Kalman estándar, sin embargo este método presenta algunos inconvenientes, de los cuales nace la necesidad de buscar otras alternativas para estimar estados y parámetros de sistemas no lineales.

Una alternativa al filtro de kalman extendido la da el conjunto de filtros de Kalman, el cual expresa el error de estimación o estadístico a partir de un conjunto de modelos, empleando este método no se tiene la necesidad de linealizar el modelo del sistema, ni tampoco de propagar en el tiempo la covarianza de estimación. A lo largo del proyecto se presentan ejemplos mediante los cuales se puede observar y comprobar el funcionamiento óptimo de este método, además es posible comprobar que el conjunto de filtros de Kalman proporciona una estimación más óptima que la proporcionada por el filtro de Kalman extendido. Finalmente empleando uno de los algoritmos desarrollados en el proyecto (conjunto de filtros de Kalman para estimación de estados) se controla un sistema de generación eólica, lo anterior se realiza para demostrar que es necesario realizar una estimación óptima si se busca diseñar un controlador a partir de la estimación.

TABLA DE CONTENIDO

1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Justificación	2
1.3. Objetivos	3
1.3.1. General	3
1.3.2. Especificos	3
1.4. Marco conceptual	4
1.5. Estado del arte	5
1.6. Alcance	6
1.7. Estructura del trabajo de grado	6
2. Filtro de Kalman	7
2.1. Estimación de estados	7
2.1.1. El problema de la estimación de estados	8
2.1.2. Ecuaciones para estimación de estados	8
2.1.3. Algoritmo para estimación de estados	10
2.1.4. Ejemplo 1. Estimación de estados	11
2.1.5. Ejemplo 2. Estimación de estados	15
2.2. Estimación de parámetros	19
2.2.1. El problema de la estimación de parámetros	19
2.2.2. Ecuaciones para estimación de parámetros	20
2.2.3. Algoritmo para estimación de parámetros	21
2.2.4. Ejemplo 1. Estimación de parámetros	22
2.2.5. Ejemplo 2. Estimación de parámetros	26
2.3. Filtro de Kalman dual	29
2.3.1. El problema de la estimación dual	29
2.3.2. Ecuaciones para estimación dual	30
2.3.3. Algoritmo para estimación dual	31
2.3.4. Ejemplo. Estimación dual	32

3. Filtro de Kalman Extendido	33
3.1. Estimación de estados	33
3.1.1. El problema de la estimación de estados	33
3.1.2. Ecuaciones para estimación de estados	34
3.1.3. Algoritmo para estimación de estados	34
3.1.4. Ejemplo 1. Estimación de estados	36
3.1.5. Ejemplo 2. Estimación de estados	41
3.2. Estimación de parámetros	45
3.2.1. El problema de la estimación de parámetros	45
3.2.2. Ecuaciones para estimación de parámetros	45
3.2.3. Algoritmo para estimación de parámetros	46
3.2.4. Ejemplo. Estimación de parámetros	48
3.3. Filtro de Kalman extendido dual	51
3.3.1. El problema de la estimación dual	52
3.3.2. Ecuaciones para estimación dual	53
3.3.3. Algoritmo para estimación dual	54
3.3.4. Ejemplo. Estimación dual	56
4. Conjunto de Filtros de Kalman	62
4.1. Estimación de estados	63
4.1.1. El problema de la estimación de estados	63
4.1.2. Deducción de ecuaciones	63
4.1.3. Ecuaciones para estimación de estados	65
4.1.4. Algoritmo para estimación de estados	66
4.1.5. Ejemplo 1. Estimación de estados	69
4.1.6. Ejemplo 2. Estimación de estados	74
4.2. Estimación de parámetros	78
4.2.1. El problema de la estimación de parámetros	78
4.2.2. Ecuaciones para estimación de parámetros	78
4.2.3. Algoritmo para estimación de parámetros	79
4.2.4. Ejemplo. Estimación de parámetros	82
4.3. Conjunto de filtros de Kalman dual	86
4.3.1. El problema de la estimación dual	86
4.3.2. Ecuaciones para estimación dual	86
4.3.3. Algoritmo para estimación dual	86
4.3.4. Ejemplo. Estimación dual	89

5. Control de un sistema de generación eólica a partir del conjunto de filtros de Kalman	95
5.1. Modelo del sistema eólico	96
5.2. Algoritmo para estimación de estados	96
5.3. Control del sistema eólico	98
5.3.1. Algoritmo para el control del sistema eólico	99
6. Análisis comparativo entre el filtro de Kalman extendido y el conjunto de filtros de Kalman	103
6.1. Estimación de estados	103
6.1.1. Estimación de estados. Ejemplo 1	103
6.1.2. Estimación de estados. Ejemplo 2	106
6.2. Estimación de parámetros	108
6.2.1. Estimación de parámetros. Ejemplo	108
7. Conclusiones	110
7.1. Futuros trabajos de investigación	112
8. Bibliografía	113

1. INTRODUCCIÓN

1.1. Planteamiento del problema

Identificar un sistema dinámico es construir un modelo matemático del sistema a partir de datos de entrada y salida observados, esta temática es muy extensa debido a que es posible implementar diferentes técnicas de identificación dependiendo de las características de los modelos a estimar [1], se distinguen dos clases de métodos, en primer lugar se encuentran los métodos tradicionales, los cuales son empleados para la identificación de sistemas lineales, y en segundo lugar se encuentran los métodos modernos, los cuales son empleados para la identificación de sistemas no lineales [2].

Entre los métodos de identificación tradicionales se encuentra el método de mínimos cuadrados, método de corrección de gradiente, método de predicción de error, método de regresión pseudo-lineal y método de máxima verosimilitud [3] [4], los métodos tradicionales pueden realizar una identificación óptima de sistemas lineales, pero no pueden ser empleados para sistemas no lineales [2], partiendo de la necesidad de identificar sistemas no lineales, en los últimos años se han desarrollado nuevos métodos con diversas características [5].

Entre los métodos modernos de identificación más relevantes se encuentran los basados en algoritmos genéticos, lógica difusa, redes neuronales, algoritmos de optimización mediante enjambre de partículas (PSO), algoritmos por modelos auxiliares, algoritmos de identificación multi-innovación, filtros de Kalman (KF), filtro de Kalman extendido (EKF) y algoritmos de identificación jerárquica [2] [6] [9] [10], a pesar del gran avance que han representado estos métodos en la identificación de sistemas dinámicos siguen existiendo grandes inconvenientes, como la dependencia de una estimación a priori del sistema, falta de sensibilidad respecto a las variaciones del sistema, poca universalidad de los métodos, y la no equivalencia total del modelo identificado frente al sistema real [2][7].

Emplear métodos de identificación con los inconvenientes mencionados anteriormente no es adecuado, ya que estos inconvenientes tienen una repercusión directa en el control del sistema dinámico, esto se debe a que el modelo construido a partir de los métodos de identificación tiene como propósito describir fielmente el comportamiento del sistema para posteriormente ser utilizado en el diseño de un controlador, por ende es necesario identificar el sistema a controlar de una manera óptima para que el controlador funcione adecuadamente [7]. Se enfrenta el gran reto de desarrollar un nuevo método de identificación que no requiera de un conocimiento a priori del sistema, que sea versátil, sensible a las variaciones, computacionalmente eficiente, y que construya un modelo matemático lo más parecido posible al sistema real, ya que es imposible la equivalencia total del modelo construido frente al sistema real [2][7].

1.2. Justificación

Los métodos de identificación son ampliamente aplicados en diversas áreas de control automático, aviación, vuelos espaciales, astronomía, medicina, economía, biología, entre otros. Con el desarrollo de la ciencia y de la tecnología establecer un modelo matemático de un sistema a partir de su respectiva identificación se ha convertido en un tema de suma importancia y utilidad [2].

El problema de encontrar un método de identificación más óptimo que los métodos implementados actualmente es un problema interesante y complicado a la vez, interesante debido a que un método más eficiente podría abordar de manera satisfactoria inconvenientes que aún no tienen solución en diversas aplicaciones industriales [1], además es complicado en primer lugar debido a que es imposible la equivalencia total del modelo construido frente al sistema real [7], y en segundo lugar debido a que una vez realizada la identificación de un sistema se busca diseñar un controlador no lineal, sí el sistema tiene un comportamiento no lineal, sin embargo hoy en día la gran mayoría de los controladores empleados en la industria son lineales puros, ya que el diseño de controladores no lineales representa grandes dificultades [5].

La identificación de sistemas no lineales se considera un problema difícil puesto que identificar un sistema no lineal conlleva dos grandes etapas: la selección de la estructura del modelo con determinado número de parámetros, y la selección de un algoritmo que estime dichos parámetros [5].

En este proyecto se propone la construcción de un método de identificación de sistemas basado en el conjunto de filtros de Kalman (EnKF), para posteriormente ser implementado en el control de una planta de generación eólica, este método es una alternativa al tradicional filtro de Kalman (KF) y representa una gran mejora respecto al manejo de espacio de estados extensos y al manejo de la evolución de error no lineal [8], mediante este proyecto se busca que al identificar un sistema y al diseñar un controlador a partir de este método novedoso se obtengan resultados más óptimos que los resultados obtenidos a partir de métodos implementados anteriormente puesto que en comparación con el tradicional filtro de Kalman (KF) y el filtro de kalman extendido (EKF); el conjunto de filtros de kalman (EnKF) representa mayor eficiencia computacional, debido a que este método supera inconvenientes relacionados con el error de la matriz de covarianza, y la identificación de sistemas no lineales [8].

1.3. Objetivos

1.3.1. General

Identificar y controlar una turbina de generación eólica a pequeña escala empleando el conjunto de filtros de Kalman (EnKF) y realizar el análisis comparativo respecto a otros métodos.

1.3.2. Especificos

- Desarrollar un método de identificación de sistemas basado en el conjunto de filtros de Kalman (EnKF)
- A partir del conjunto de filtros de Kalman controlar un sistema de generación eólica

1.4. Marco conceptual

Identificación: diseño de una señal de entrada apropiada usando datos de entrada y salida experimentales, selección de una clase de modelos, construcción de una función de criterio de error, determinación de un modelo que se ajuste mejor a los datos mediante métodos de optimización.

Estado: conjunto mínimo de datos que describen el comportamiento dinámico de un sistema.

Modelo: relación entre cantidades observadas, la cual se expresa generalmente mediante una expresión matemática, tiene como finalidad predecir el comportamiento de un objeto.

Clase de modelos: conjunto o colección de modelos, puede ser un conjunto parametrizado por parámetros de dimensión finita. Ejemplo: todos los modelos de espacio de estado lineales de orden n .

Espacio de estados: una representación de espacios de estados es un modelo matemático de un sistema físico descrito mediante un conjunto de entradas, salidas y variables de estado relacionadas por ecuaciones diferenciales de primer orden que se combinan en una ecuación diferencial matricial de primer orden.

Controlador: elemento encargado de manipular la entrada al sistema para obtener el efecto deseado en la salida del sistema (retroalimentación).

Filtro: Es un elemento que deja pasar señales eléctricas a través de él, a una determinada frecuencia o rangos de frecuencia mientras previene el paso de otras, pudiendo modificar tanto su magnitud como fase.

Control robusto: Es la rama de la ingeniería de control encargada del diseño de controladores estables y robustos en presencia de errores de modelado.

Estocástico: se denomina estocástico al sistema cuyo comportamiento es intrínsecamente no determinista.

1.5. Estado del arte

En 1965 se realizaron los avances más significativos al desarrollo de la teoría de identificación de sistemas. Tanto los aportes de Ho y Kalman, como los aportes de Åström y Bohlin abrieron paso a la teoría de identificación moderna, además se dio origen a las dos corrientes de investigación más importantes en el campo de la teoría de identificación (identificación de sub espacios y predicción de error) [14] [15] [16].

Hasta finales de 1950, gran parte del diseño de controladores estaba basado en Bode, Nyquist, gráficos Ziegler-Nichols o análisis de respuesta al impulso, estas técnicas estaban limitadas a diseñar controladores para sistemas con una sola entrada y una sola salida; estos sistemas se conocen como SISO (single input, single output) [14]. En 1960 Kalman introduce la representación en espacio de estados y además puso las bases para la teoría de espacio de estados basado en filtrado óptimo y control óptimo mediante la implementación del filtro de Kalman [16][22].

El diseño de controladores se limitaba a los campos relacionados con aplicaciones mecánicas, eléctricas y aeroespaciales; sin embargo, Los aportes de Kalman representaron el desarrollo de modelos dinámicos mediante los cuales fue posible el desarrollo de controladores para campos relacionados con procesos de control, sistemas ambientales, sistemas biológicos, sistemas biomédicos, entre otros [14] [16]. Las investigaciones de Ho y Kalman acerca de la representación en espacio de estados habían encontrado la solución de problemas determinísticos; sin embargo, de En 1974 Akaike encontraría la solución de problemas estocásticos mediante la identificación de sub-espacios [14] [17].

Por otra parte Åström y Bohlin introdujeron a la comunidad de control el modelo ARMA y ARMAX, además del marco de máxima verosimilitud. Estos aportes darían paso al desarrollo de la predicción de error en la teoría de identificación [14] [15]. En 1978 los investigadores pasaron de buscar “el sistema verdadero” a la “mejor aproximación del sistema” [14]. Finalmente en 1986 Gevers and Ljung, con su libro “System identification: Theory for the user” darían lugar al reconocimiento de la teoría de identificación como un problema de diseño [14] [7]. Hoy en día el campo de la identificación de sistemas abarca áreas como el diseño experimental, la identificación de bucle cerrado, identificación en el dominio de la frecuencia, estimación de incertidumbre, y control robusto [14].

En el presente proyecto se tiene como referente “Modelling and Control of a Small Scale Wind Turbine” [18]. En la citada investigación se obtuvieron inconvenientes rela-

cionados con el diseño de controladores debido a que se utilizaron métodos de control inadecuados para sistemas no lineales (controlador PID y control óptimo LQR (linear quadratic regulator)); por lo tanto, en este proyecto se busca implementar un método adecuado para sistemas no lineales capaz de identificar el sistema de manera óptima para posteriormente diseñar los controladores correspondientes.

1.6. Alcance

La investigación se va a centrar en la metodología de solución, debido a que se implementara una metodología nueva a un problema clásico. La base teórica de la metodología a implementar se fundamenta en el conjunto de filtros de Kalman (EnKF), y el problema clásico es la identificación óptima de un sistema de generación eólica para posteriormente diseñar los controladores del sistema.

1.7. Estructura del trabajo de grado

El trabajo de grado está organizado de la siguiente forma. En el capítulo 2 se aborda el problema de la estimación dual a partir del filtro de Kalman, en el capítulo 3 y 4 se aborda la misma problemática a partir del filtro de Kalman extendido y el conjunto de filtros de Kalman respectivamente, en el capítulo 5 se desarrolla un controlador por realimentación de estados a partir de la estimación de estados realizada por el conjunto de filtros de Kalman, en el capítulo 6 se realiza un análisis comparativo entre el filtro de Kalman extendido y el conjunto de filtros de Kalman, finalmente se presentan las conclusiones en el capítulo 7.

2. FILTRO DE KALMAN

En 1960 R. E. Kalman presenta por primera vez el filtro de Kalman como solución al problema de filtrado lineal óptimo, desde entonces este algoritmo es objeto de diversas investigaciones y aplicaciones, principalmente en el área de navegación asistida [23].

El filtro de Kalman tiene su origen en la formulación de espacio de estados para sistemas dinámicos lineales, proporciona una solución recursiva al problema de filtrado óptimo lineal, debido a que para calcular la estimación actual de estados se utiliza la estimación anterior y los nuevos datos de entrada, además es considerado como un algoritmo de procesamiento de datos óptimo, debido a que minimiza el error cuadrático medio (diferencia entre estimación y lo que se estima). En la práctica, es posible que las variables de estado de un sistema dinámico no puedan ser determinadas por una medición directa, dado lo anterior su medición se realiza por medio de procesos estocásticos que involucran un grado de incertidumbre en la medición [19][23][26].

2.1. Estimación de estados

El filtro de Kalman aborda el problema de la estimación de estados, para un momento determinado se realiza una predicción de estado, la cual es corregida o mejorada mediante un tipo de control feedback (retroalimentación) representada por mediciones realizadas al sistema [23].

Una descripción más detallada acerca del filtro de Kalman puede encontrarse en [19], [23] y [26], estas investigaciones se toman como referencia para definir el problema y las ecuaciones correspondientes a la estimación de estados.

2.1.1. El problema de la estimación de estados

- Modelo en espacio de estados

El filtro de Kalman aborda el problema de estimar los estados de un sistema que evoluciona respecto a las siguientes ecuaciones diferenciales.

$$x_{k+1} = F_{k+1,k}x_k + w_k \quad (2.1)$$

Donde F se conoce como matriz de transición de estados, y x se conoce como vector de estados.

$$z_k = H_k x_k + v_k \quad (2.2)$$

Donde H se conoce como matriz de medición, y z o z_m se conoce como vector de observación o salida medida del sistema.

Además w_k y v_k corresponden al ruido del proceso y al ruido de la medición respectivamente, los ruidos se asumen aditivos, blancos y gaussianos, con media cero y con matriz de covarianza Q_k y R_k respectivamente.

2.1.2. Ecuaciones para estimación de estados

Las ecuaciones del filtro de Kalman para estimación de estados son las siguientes.

- Inicialización

$$x = E[x] \quad (2.3)$$

$$P = E[(x - E[x])(x - E[x])^T] \quad (2.4)$$

- Estimación de estados a-priori

$$\underline{x}_k = F_{k+1,k}x \quad (2.5)$$

- Covarianza de estimación a-priori

$$\underline{P}_k = F_{k+1,k}P_{k-1}F_{k+1,k}^T + Q_{k-1} \quad (2.6)$$

- Ganancia de Kalman

$$G_k = \underline{P}_k H_k^T [H_k \underline{P}_k H_k^T + R_k]^{-1} \quad (2.7)$$

- Estimación de estados a-posteriori

$$\underline{x}_k = \underline{x}_k + G_k(z_{m_k} - H_k \underline{x}_k) \quad (2.8)$$

- Covarianza de estimación a-posteriori

$$P_k = (I - G_k H_k) P_k \quad (2.9)$$

En la figura 2.1, se observa el esquema del filtro de Kalman, en el cual se puede apreciar tanto el ruido correspondiente al proceso (evolución de estado) como el ruido correspondiente a la medición, además se puede apreciar la salida real del sistema zr , la salida medida del sistema denotada por z o zm , en la cual se tiene en cuenta el ruido de la medición y finalmente la salida estimada del sistema ze , la cual es igual al producto entre la matriz de medición H y el estado estimado a-posteriori \hat{x} .

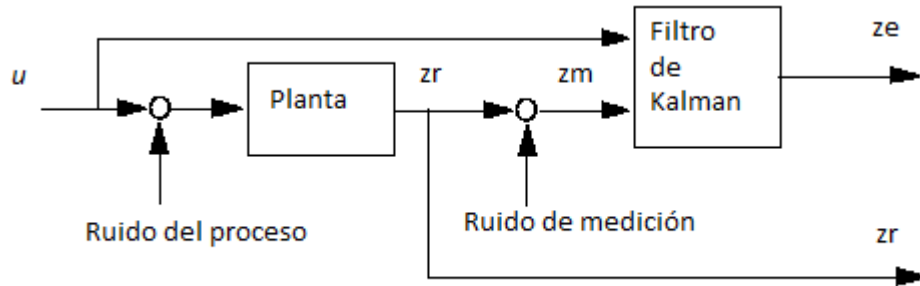


Figura 2.1: Salidas del filtro de Kalman

2.1.3. Algoritmo para estimación de estados

Las ecuaciones del filtro de kalman para la estimación de estados se implementan en el algoritmo 1.

Algoritmo 1 Filtro de Kalman para estimación de estados

Require: $A, x, P, H, zm, Q, R, u, B$

Ensure: x, P

function $[x, P] = \text{filtro de Kalman-estados}(A, x, P, H, zm, Q, R, u, B)$

Predicción

$\underline{x} = Ax + Bu // \text{estimación de estado a-priori}$

$\underline{P} = APA^T + Q // \text{covarianza de estimación a-priori}$

Corrección

$G = \underline{P}H^T[H\underline{P}H^T + R]^{-1} // \text{ganancia de Kalman}$

$x = \underline{x} + G(zm - H\underline{x}) // \text{estimación de estado a-posteriori}$

$P = (I - GH)\underline{P} // \text{covarianza de estimación a-posteriori}$

return

El algoritmo 1 (filtro de Kalman para estimación de estados), tiene como objetivo estimar los estados de un sistema lineal, es posible observar que las entradas del algoritmo son: la matriz de transición de estados denotada por F o A , el estado estimado inicial x , la covarianza de estimación de estados inicial P , la matriz de medición H , la salida medida del sistema zm , las covarianzas del proceso y medición respectivamente Q, R , además al modelo en espacio de estados del sistema se incluye la matriz de entrada B y la señal de control u .

Una vez el algoritmo reconoce las entradas mencionadas anteriormente, se realiza la estimación a-priori de estados y de covarianza (predicción), posteriormente se calcula la ganancia de Kalman, y finalmente se realiza la estimación a-posteriori de estados y covarianza (corrección), las estimaciones a-posteriori son las salidas del algoritmo.

El filtro de Kalman se considera un algoritmo de solución recursiva debido a que el estado estimado x y la covarianza de estimación de estado P calculados en una iteración se emplean en la próxima iteración como estimación de estado inicial y covarianza de estimación de estado inicial.

2.1.4. Ejemplo 1. Estimación de estados

Se presenta un ejemplo (algoritmo 2) tomado de la herramienta de MathWorks “learning Kalman filter” en el cual se implementa el algoritmo 1 correspondiente al filtro de Kalman para estimación de estados, el ejemplo describe la medida de un voltímetro en un automóvil.

Se puede observar en el algoritmo 2 que en primer lugar se introducen las entradas que requiere el filtro de kalman A, x, P, H, Q, R, u, B con excepción de la salida medida zm , la cual se calcula en el proceso iterativo para cada muestra, se define el orden del sistema n , el número de iteraciones N que se realizan en el algoritmo, y cuatro matrices de almacenamiento: zmV almacenamiento de mediciones o salida medida (zm), zeV almacenamiento de la salida estimada ($ze = Hx$), xV almacenamiento de la estimación de estados (x), y finalmente $truV$ almacenamiento del estado real del sistema (tru).

A continuación se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones N indicado, luego se genera un voltaje aleatorio tru , en el cual se tiene en cuenta el ruido del proceso o de evolución de estado, tru se almacena en la matriz $truV$, a continuación se define la salida medida zm , la cual tiene en cuenta el ruido de la medición o del sensor, zm se almacena en la matriz zmV , como paso siguiente se realiza una iteración del filtro de Kalman, por ende como se explicó anteriormente se obtiene la estimación de estados x , y la covarianza de estimación P , las cuales actualizan los valores iniciales de estimación, x es almacenada en la matriz xV , finalmente se calcula la salida estimada ze multiplicando la matriz de medición H por el estado estimado x , ze se almacena en la matriz zeV .

Como paso siguiente se define la salida real del sistema zr , en la figura 2.1 se observa que esta salida es igual a la salida medida pero sin tener en cuenta el ruido de medición, además se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y sus respectivas covarianzas de error $MeasErrCov$ y $EstErrCov$.

Una vez las iteraciones han terminado, las matrices de almacenamiento $truV$ y xV contienen todos los voltajes verdaderos y todos los estados o voltajes estimados respectivamente, las matrices mencionadas son de tamaño $(1 \times N)$, debido a que el sistema es de primer orden, en la figura 2.2 se observa la matriz $truV$ (color verde) y la matriz xV (color azul), es posible deducir que los estados estimados se aproximan de manera considerable a los estados verdaderos del sistema.

Algoritmo 2 Ejemplo 1. Filtro de Kalman para estimación de estados. Voltímetro de un automóvil

```
Voltímetro de un automóvil. medida= 12v
n = 1//orden del sistema
A = 1//matriz de transición de estados
Q = 22 //ruido del proceso (stdev=2v)
H = 1// matriz de medición
R = 22 //ruido de la medición (stdev=2v)
B = 0//matriz de entrada
u = 12//señal de control
x = 12//estimación de estado inicial
P = 22//covarianza de estimación de estado inicial
iteraciones
N = 80
matrices de almacenamiento
zmV = zeros(1, N)//almacenamiento de mediciones o salida medida (zm)
zeV = zeros(1, N)//almacenamiento de la salida estimada (Hx=ze)
xV = zeros(n, N)//almacenamiento de la estimación de estados (x)
truV = zeros(n, N)//almacenamiento del estado real del sistema (tru)
proceso iterativo
for k = 1 : N do
    tru = randn2 + 12//generación de voltajes aleatorios
    truV(1, k) = tru //almacenamiento de los voltajes
    zm = tru + randn2//medición de los voltajes
    zmV(1, k) = zm//almacenamiento de las mediciones
    [x, P] = filtro de Kalman-estados(A, x, P, H, zm, Q, R, u, B)//realizar una iteración
    del filtro de Kalman
    xV(1, k) = x //almacenamiento de la estimación de estados
    zeV(1, k) = Hx//almacenamiento de la salida o mediciones estimadas
end for
cálculo de errores
zr = truV//salida real del sistema
MeasErr = zr - zmV//error de medición
EstErr = zr - zeV//error de estimación
MeasErrCov = sum(MeasErrMeasErr)/length(MeasErr)//covarianza del error
de medición
EstErrCov = sum(EstErrEstErr)/length(EstErr)//covarianza del error de esti-
mación
```

De igual manera si las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas medidas y todas las salidas estimadas respectivamente, las matrices mencionadas son de tamaño $1 \times N$, debido a que el sistema tiene una sola salida, en la figura 2.3 se observa la matriz zmV (color rojo, puntos) y la matriz zeV (color magenta), es posible deducir que las salidas estimadas se aproximan de manera considerable a las salidas medidas del sistema.

Finalmente en la figura 2.4 se observa la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y entre el error de estimación $EstErr$ (color azul), además la figura muestra que el nivel de ruido es reducido una vez se ha implementado el filtro de Kalman, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, la covarianza de error antes de la implementación del filtro ($MeasErrCov$) es igual a 3,1178 y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 2,5670.

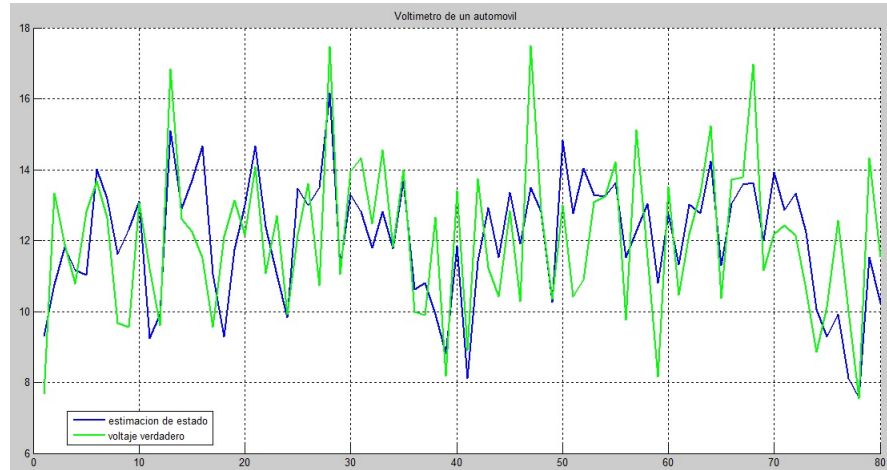


Figura 2.2: Estimación de estados mediante el filtro de Kalman. Ejemplo 1 (voltímetro de un automóvil)

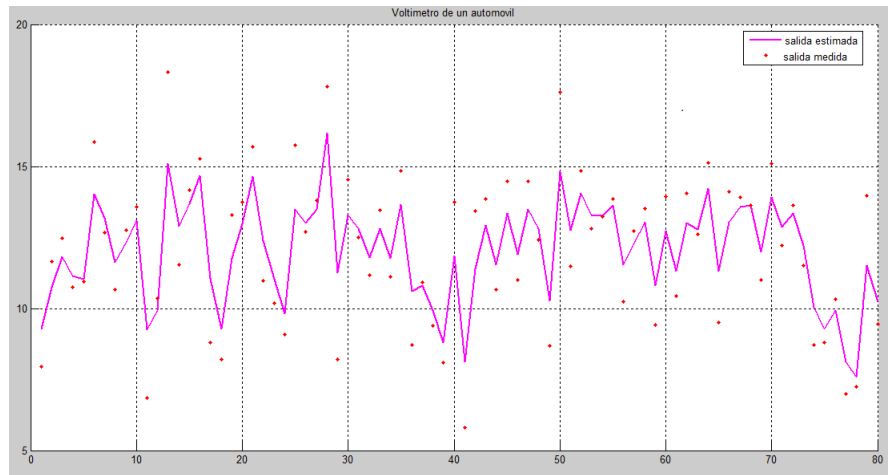


Figura 2.3: Salida estimada mediante el filtro de Kalman. Ejemplo 1 (voltímetro de un automóvil)

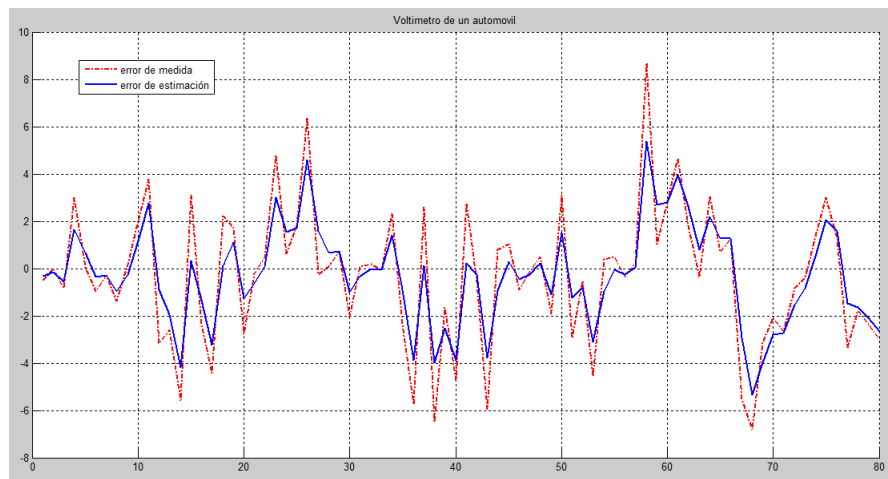


Figura 2.4: Error de estimación mediante el filtro de Kalman. Ejemplo 1 (voltímetro de un automóvil)

2.1.5. Ejemplo 2. Estimación de estados

Se presenta un ejemplo (algoritmo 3,4) en el cual se implementa el algoritmo 1 correspondiente al filtro de Kalman para estimación de estados, el ejemplo describe un sistema SISO (una entrada, una salida) de segundo orden. Mediante la herramienta de Matlab “drss” se genera un modelo en espacio de estados en tiempo discreto, de esta manera se obtiene la matriz de transición de estados A , la matriz de entrada B y la matriz de medición H .

Se puede observar en el algoritmo 2 que en primer lugar se introducen las entradas que requiere el filtro de Kalman con excepción de la salida medida zm , se define el número de iteraciones N que se realizarán en el algoritmo, y cinco matrices de almacenamiento: zrV almacenamiento de la salida real del sistema, zmV almacenamiento de mediciones o salida medida (zm), zeV almacenamiento de la salida estimada ($Hx=ze$), xV almacenamiento de la estimación de estados (x), y finalmente $truV$ almacenamiento del estado real del sistema (tru).

A continuación se da paso al proceso iterativo, en el cual se describe la evolución del estado real del sistema de acuerdo con la ecuación 2.1, se prosigue con el almacenamiento del estado real del sistema en la matriz $truV$, luego se calcula la salida medida del sistema zm , la cual se describe mediante la ecuación 2.2, se prosigue con el almacenamiento de dicha medición en la matriz zmV , después se calcula la salida real del sistema zr , la cual es igual a la salida medida sin tener en cuenta el ruido de la medición, esta salida se almacena en zrV .

Como paso siguiente se realiza una iteración del filtro de Kalman, por ende como se explicó anteriormente se obtiene la estimación de estados x , y la covarianza de estimación de estado P , las estimaciones mencionadas se utilizan en la siguiente iteración como estimaciones iniciales, x es almacenada en la matriz xV , además se calcula el producto entre la matriz de medición H y x conocido como salida estimada ze , la cual se almacena en la matriz zeV , como último paso se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y las covarianzas de error $MeasErrCov$ y $EstErrCov$.

Una vez las iteraciones han terminado, las matrices de almacenamiento $truV$ y xV contienen todos los estados verdaderos y todos los estados estimados respectivamente, a diferencia del ejemplo 1, en el ejemplo 2 las matrices $truV$ y xV son matrices de tamaño $(2 \times N)$ debido a que se tienen dos estados reales y dos estados estimados.

Algoritmo 3 Ejemplo 2. Filtro de Kalman para estimación de estados. Sistema SISO de segundo orden

```

n = 2//orden del sistema
A=  $\begin{bmatrix} 0,04867 & 0,804 \\ 0,804 & 0,07916 \end{bmatrix}$ //matriz de transición de estados
Q = diag([0,520,22)]//ruido del proceso
H =  $\begin{bmatrix} -0,888 & 0,1001 \end{bmatrix}$  //matriz de medición
R = 0,1//ruido de la medición
B = [-1,795;0,8404]//matriz de entrada
u = 0,1randn//señal de control
x = 0,1rand(n,1)//estimación de estado inicial
P = 10eye(n)//covarianza de estimación de estado inicial
iteraciones
N = 100
matrices de almacenamiento
zrV = zeros(1, N)//almacenamiento de la salida real (zr)
zmV = zeros(1, N)//almacenamiento de mediciones o salida medida (zm)
zeV = zeros(1, N)//almacenamiento de la salida estimada (Hx=ze)
xV = zeros(n, N)//almacenamiento de la estimación de estados (x)
truV = zeros(n, N)//almacenamiento del estado real del sistema (tru)
estado real inicial
tru = zeros(n, 1)
proceso iterativo
for k = 1 : N do
    tru = Atru + Bu + sqrt(Q)randn(n,1)//estado real del sistema
    truV(:, k) = tru //almacenamiento del estado real del sistema
    zm = Htru + sqrt(R)randn//medición de los voltajes
    zmV(1, k) = zm//almacenamiento de las mediciones
    zr = Htru//salida real del sistema
    zrV(1, k) = zr//almacenamiento de las salidas reales del sistema
    [x, P] = filtro de Kalman-estados(A, x, P, H, zm, Q, R, u, B)//realizar una iteración
    del filtro de Kalman
    xV(:, k) = x //almacenamiento de la estimación de estados
    zeV(:, k) = Hx//almacenamiento de la salida o mediciones estimadas
end for

```

Algoritmo 4 Ejemplo 2. Filtro de Kalman para estimación de estados. Sistema SISO de segundo orden, continuación

cálculo de errores

$MeasErr = zr - zmV$ //error de medición

$EstErr = zr - zeV$ //error de estimación

$MeasErrCov = sum(MeasErrMeasErr)/length(MeasErr)$ //covarianza del error de medición

$EstErrCov = sum(EstErrEstErr)/length(EstErr)$ //covarianza del error de estimación

En la figura 2.5 se observa la matriz $truV$ (color verde) y la matriz xV (color azul), es posible deducir que los estados estimados coinciden casi por completo con los estados verdaderos del sistema, de igual manera una vez las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas medidas y todas las salidas estimadas respectivamente, al igual que en el ejemplo 1, en el ejemplo 2 las matrices mencionadas son matrices de tamaño $(1 \times N)$ debido a que el sistema tiene una sola salida, por ende la salida medida y estimada es la misma para los cuatro estados del sistema, se puede observar en la figura 2.6 la matriz zmV (color rojo, puntos) y zeV (color magenta), las salidas estimadas coinciden casi por completo con las salidas verdaderas.

Finalmente en la figura 2.7 se puede observar la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y entre el error de estimación $EstErr$ (color azul), la figura muestra que el nivel de ruido es reducido una vez se ha implementado el filtro de Kalman, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, la covarianza de error antes de la implementación del filtro ($MeasErrCov$) es igual a 0,1134 y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 0,0768.

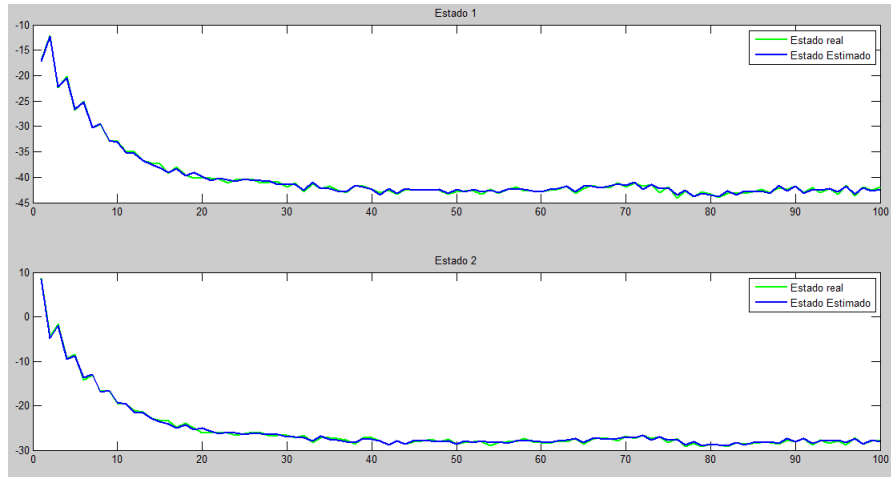


Figura 2.5: Estimación de estados mediante el filtro de Kalman. Ejemplo 2 (sistema SISO de segundo orden)

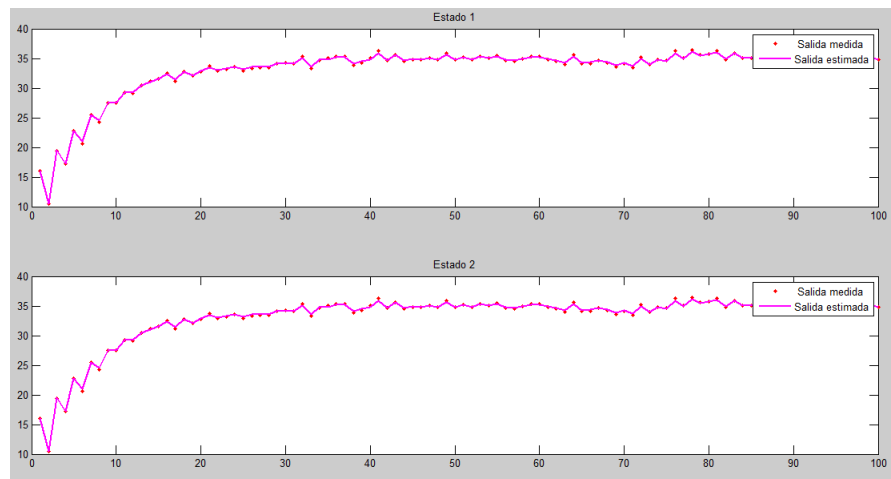


Figura 2.6: Salidas estimadas mediante el filtro de Kalman. Ejemplo 2 (sistema SISO de segundo orden)

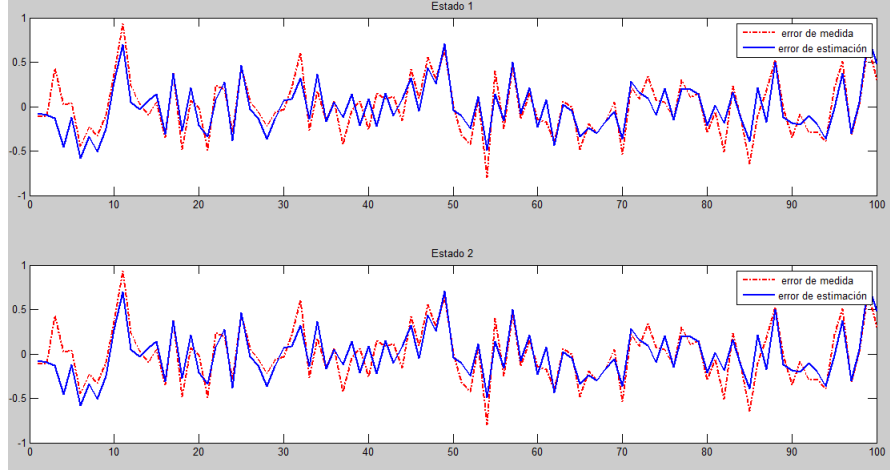


Figura 2.7: Error de estimación mediante el filtro de Kalman. Ejemplo 2 (sistema SISO de segundo orden)

2.2. Estimación de parámetros

El filtro de Kalman también puede emplearse para estimar los parámetros de un sistema, los cuales se obtienen a partir de la función de transferencia del mismo, los parámetros estimados en ocasiones pueden ser empleados para construir el modelo del sistema [7][25].

Una descripción más detallada acerca de identificación de sistemas o estimación de parámetros se encuentran en [7],[19] y [27], estas investigaciones se toman como referencia para definir el problema y las ecuaciones correspondientes a la estimación de parámetros mediante el filtro de Kalman.

2.2.1. El problema de la estimación de parámetros

Para estimar los parámetros de un sistema, se aborda el problema de estimación recursiva, considerando el siguiente modelo de regresión lineal.

$$\theta_k = \theta_{k-1} + w_k \quad (2.10)$$

$$y_k = \phi_k^T \theta_k + v_k \quad (2.11)$$

Donde y_k se conoce como el observador de parámetros, el vector θ contiene los verdaderos parámetros del sistema, y el vector de regresión ϕ contiene salidas y entradas anteriores.

$$\phi_k^T = C_k = (y_{k-1}, \dots, y_{k-n}, u_{k-1}, \dots, u_{k-m}) \quad (2.12)$$

Se obtiene una nueva representación en espacio de estados, donde θ_k corresponde a un proceso estacionario con matriz de transición de estados igual a la matriz identidad, además para procesos estacionarios es posible despreciar los ruidos de proceso y medición w_k y v_k .

2.2.2. Ecuaciones para estimación de parámetros

- Inicialización

$$\theta = E[\theta] \quad (2.13)$$

$$P_\theta = E[(\theta - E[\theta])(\theta - E[\theta])^T] \quad (2.14)$$

- Estimación de parámetros a-priori

$$\underline{\theta}_k = \theta_{k-1} \quad (2.15)$$

- Covarianza de estimación a-priori

$$\underline{P}_k = P_{k-1} + Q_{k-1} \quad (2.16)$$

- Ganancia de Kalman

$$G_k = \underline{P}_k C_k^T [C_k \underline{P}_k C_k^T + R_k]^{-1} \quad (2.17)$$

- Estimación de parámetros a-posteriori

$$\theta_k = \underline{\theta}_k + G_k (y_k - C_k \underline{\theta}_k) \quad (2.18)$$

- Covarianza de estimación a-posteriori

$$P_k = (I - G_k C_k) \underline{P}_k \quad (2.19)$$

2.2.3. Algoritmo para estimación de parámetros

Las ecuaciones del filtro de Kalman para estimación de parámetros se implementan en el algoritmo 5.

Algoritmo 5 Filtro de Kalman para estimación de parámetros

Require: θ, P, C, y, Q, R

Ensure: θ, P

function $[x, P] = \text{filtro de Kalman-parámetros}(\theta, P, C, y, Q, R)$

Predicción

$\underline{\theta} = \theta$ // estimación de parámetros a-priori

$\underline{P} = P + Q$ // covarianza de estimación a-priori

Corrección

$G = \underline{P}C^T[C\underline{P}C^T + R]^{-1}$ // ganancia de Kalman

$\theta = \underline{\theta} + G(y - C\underline{\theta})$ // estimación de parámetros a-posteriori

$P = (I - GC)\underline{P}$ // covarianza de estimación a-posteriori

return

El algoritmo 5 (Filtro de Kalman para estimación de parámetros), tiene como objetivo estimar los parámetros de un sistema lineal. Se puede observar que las entradas del algoritmo son: la estimación de parámetros inicial θ , covarianza de estimación inicial P , el vector de regresión C , la salida real del sistema y , y las covarianzas del proceso y medición respectivamente Q, R .

Una vez el algoritmo reconoce las entradas mencionadas anteriormente, se realiza la estimación a-priori de parámetros y de covarianza (predicción), posteriormente se calcula la ganancia de Kalman y finalmente se realiza la estimación a-posteriori de parámetros y covarianza (corrección), las estimaciones a-posteriori son las salidas del algoritmo.

Se considera al filtro de Kalman como un algoritmo de solución recursiva debido a que los parámetros estimados θ y la covarianza de estimación P calculados en una iteración son empleados en la próxima iteración como estimación de parámetros inicial y covarianza de estimación inicial.

2.2.4. Ejemplo 1. Estimación de parámetros

Se presenta un ejemplo (algoritmo 6) en el cual se implementa el algoritmo 5 correspondiente al filtro de Kalman para estimación de parámetros. El ejemplo describe un sistema SISO de primer orden, por ende consta de dos parámetros.

Se puede observar en el algoritmo 6 que en primer lugar se introducen las entradas que requiere el filtro de Kalman para estimación de parámetros (θ, P, C, Q, R) con excepción del vector de regresión C y la señal de observación de parámetros yr , los cuales se calculan para cada muestra en el proceso iterativo, se define el orden del sistema n , el número de parámetros m , los parámetros verdaderos del sistema θ_{tru} , y las inicializaciones de la salida y la entrada anterior $y1$ (y_{k-1}), $u1$ (u_{k-1}).

Como paso siguiente se define el número de iteraciones o mediciones N que se realizan en el algoritmo, luego se definen cuatro matrices de almacenamiento: yrV almacenamiento del observador de parámetros (yr), uV almacenamiento de la señal de control (u), yeV almacenamiento del observador estimado de parámetros (ye), y finalmente θV almacenamiento de la estimación de parámetros (θ).

A continuación se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N que este indicado, luego se define una señal de control u , la cual en este caso representa un valor fijo, se prosigue almacenando la señal de control, posteriormente se define el vector de regresión C el cual contiene salidas y entradas anteriores (ecuación 2.11), para la primera iteración el vector C utiliza los valores iniciales de la salida y la entrada anterior, luego se define el observador de parámetros yr (ecuación 2.10), el cual se almacena en la matriz yrV , es importante mencionar que el valor de θ_{tru} es constante, a diferencia del vector de regresión C , el cual cambia para cada iteración.

Posteriormente se realiza una iteración del filtro de Kalman para estimación de parámetros, por ende como se explicó anteriormente se obtiene la estimación de parámetros θ , y la covarianza de estimación P , las cuales se utilizan como estimación de parámetros inicial y covarianza de estimación inicial en la siguiente iteración, luego θ se almacena en la matriz θV , se prosigue calculando la señal de observación estimada de parámetros ye multiplicando el vector de regresión C por los parámetros estimados θ , finalmente ye es almacenado en yeV .

Algoritmo 6 Ejemplo 1. Filtro de Kalman para estimación de parámetros. Sistema de primer orden

```
 $n = 1$  //orden del sistema
 $m = 2$  //parámetros del sistema
 $Q = \text{diag}([0, 5^2 0, 2^2])$  //ruido del proceso
 $R = 0,1$  //ruido de la medición
 $\theta = [0; 0]$  //estimación de parámetros inicial
 $P = 100^m \text{eye}(2)$  //covarianza de estimación de parámetros inicial
 $\theta_{tru} = [0,8187; 0,1813]$  //parámetros verdaderos del sistema
 $y1 = 0$  //inicialización salida anterior
 $u1 = 0$  //inicialización entrada anterior
mediciones o iteraciones
 $N = 50$ 
matrices de almacenamiento
 $yV = \text{zeros}(1, N)$  //almacenamiento del observador de parámetros (yr)
 $uV = \text{zeros}(1, N)$  //almacenamiento de la señal de control (u)
 $yeV = \text{zeros}(1, N)$  //almacenamiento del observador estimado de parámetros (ye)
 $\theta V = \text{zeros}(m, N)$  //almacenamiento de los parámetros estimados ( $\theta$ )
proceso iterativo
for  $k = 1 : N$  do
    if  $k \geq 2$ 
         $y1 = yV(1, k - 1)$  //salida anterior
         $u1 = uV(1, k - 1)$  //entrada anterior
    end if
     $u = 30$  //señal de control
     $uV(1, k) = u$  //almacenamiento de la señal de control
     $C = [y1, u1]$  //vector de regresión
     $y = C\theta_{tru}$  //señal de observación de parámetros
     $yV(1, k) = y$  //almacenamiento del observador de parámetros
     $[x, P] = \text{filtro de Kalman-parámetros}(\theta, P, C, y, Q, R)$  //realizar una iteración del filtro de Kalman
     $\theta V(:, k) = \theta$  //almacenamiento de la estimación de parámetros
     $ye = C\theta$  //señal de observación estimada de parámetros
     $yeV(1, k) = ye$  //almacenamiento del observador estimado de parámetros
end for
```

Para la segunda iteración la salida y la entrada anterior $y1$ (y_{k-1}) y $u1$ (u_{k-1}) son actualizadas a partir de la señal de observación de parámetros yr calculada en la primera iteración, el algoritmo realiza la estimación de parámetros utilizando cada medición (observador de parámetros yr) tan pronto como esté disponible, por ende se considera que el algoritmo realiza una estimación on-line.

Es importante mencionar que la covarianza Q representa el ruido del proceso, para el caso de estimación de estados, el proceso hace referencia a la evolución de estados (ecuación 2.1), para el caso de estimación de parámetros, el proceso tiene naturaleza estacionaria, y hace referencia a los parámetros verdaderos del sistema (ecuación 2.10); por otro lado, la covarianza R representa el ruido de medición, para el caso de estimación de estados, medición hace referencia a la salida medida del sistema (ecuación 2.2), para el caso de estimación de parámetros, medición hace referencia al observador de parámetros del sistema (ecuación 2.11).

Una vez las iteraciones han terminado, la matriz de almacenamiento θV contienen todos los parámetros estimados, en la figura 2.4 es posible observar que para la iteración cinco la estimación de parámetros ya se encuentra estabilizada, como se mencionó anteriormente, en este caso se estiman dos parámetros, $a1$ (color verde) y $b0$ (color azul).

De igual manera si las iteraciones han terminado, las matrices de almacenamiento yrV y yeV contienen todas las observaciones y todas las observaciones estimadas respectivamente, es importante destacar que las matrices mencionadas son de tamaño $1 \times N$, debido a que el sistema tiene una sola salida, en la figura 2.3 se observa la matriz yrV (color rojo, puntos) y la matriz yeV (color magenta), es posible deducir que las observaciones coinciden con las observaciones estimadas del sistema.

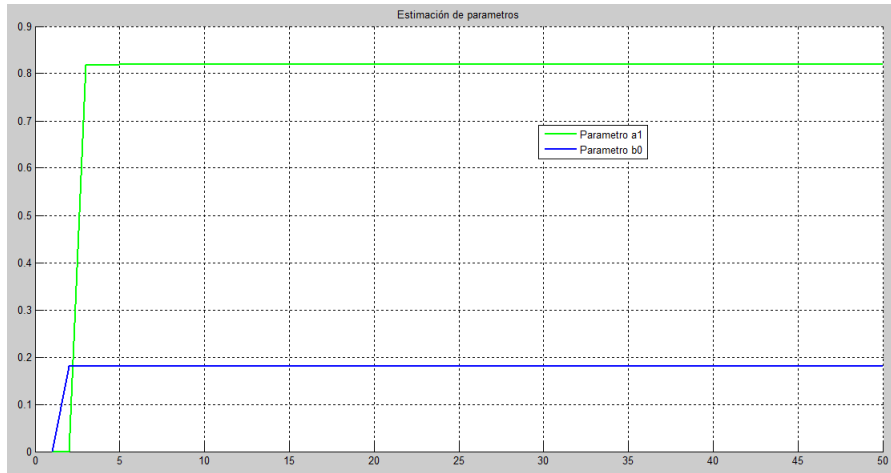


Figura 2.8: Estimación de parámetros mediante el filtro de Kalman. Ejemplo 1 (sistema de primer orden)

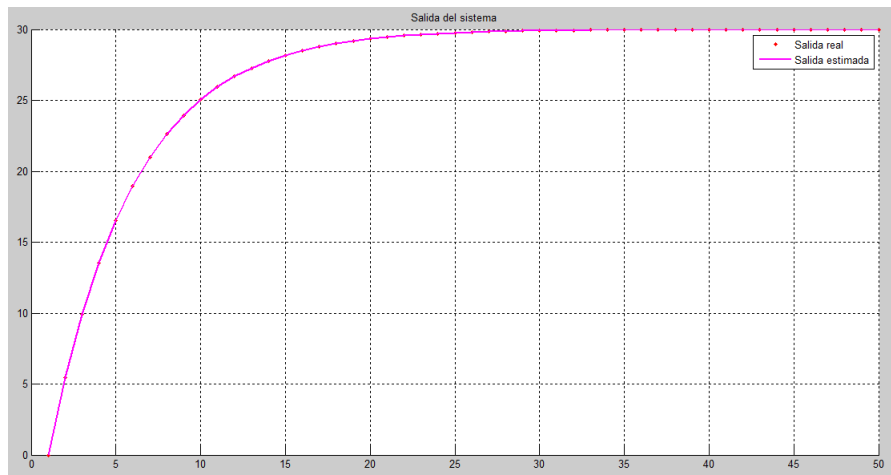


Figura 2.9: Salida estimada mediante el filtro de Kalman. Ejemplo 1 (sistema de primer orden)

2.2.5. Ejemplo 2. Estimación de parámetros

Se presenta un ejemplo (algoritmo 7) en el cual se implementa el algoritmo 5 correspondiente al filtro de Kalman para estimación de parámetros. El ejemplo describe un sistema SISO de segundo orden, por ende consta de cuatro parámetros.

Se puede observar en el algoritmo 7,8 que en primer lugar se introducen las entradas que requiere el filtro de Kalman para estimación de parámetros con excepción del vector de regresión C y la señal de observación de parámetros yr , los cuales se calculan en el proceso iterativo, se define el orden del sistema n , el número de parámetros m , los parámetros verdaderos del sistema θtru , y las inicializaciones de las salidas y la entradas anteriores $y1 (y_{k-1})$, $y2 (y_{k-2})$, $u1 (u_{k-1})$, $u2 (u_{k-2})$.

Como paso siguiente se define el número de iteraciones N , luego se definen cuatro matrices de almacenamiento. A continuación se da paso al proceso iterativo, se define una señal de control u , se prosigue almacenando la señal de control, posteriormente se define el vector de regresión C el cual contiene salidas y entradas anteriores (ecuación 2.11), para la primera iteración el vector C utiliza los valores iniciales de salidas y entradas anteriores, luego se define el observador de parámetros yr (ecuación 2.10), el cual se almacena en su respectiva matriz.

Posteriormente se realiza una iteración del filtro de Kalman para estimación de parámetros, por ende se obtiene la estimación de parámetros θ , y la covarianza de estimación P , θ se almacena en la matriz θV , se prosigue calculando la señal de observación estimada de parámetros ye , la cual se almacena en yeV , para la segunda iteración la salida y la entrada anterior $y1 (y_{k-1})$ y $u1 (u_{k-1})$ son actualizadas a partir de la señal de observación de parámetros yr calculada en la primera iteración, por otro lado la salida y la entrada anterior $y2 (y_{k-2})$ y $u2 (u_{k-2})$ son representadas por sus valores iniciales, desde la tercera iteración en adelante todas las entradas y salidas anteriores se actualizan a partir de observaciones anteriores.

Una vez las iteraciones han terminado, la matriz de almacenamiento θV contienen todos los parámetros estimados, en la figura 2.4 es posible observar que para la iteración cinco la estimación de parámetros ya se encuentra estabilizada, como se mencionó anteriormente, en este caso se estiman cuatro parámetros: $a1$ (color verde), $a2$ (color azul), $b0$ (color rojo) y $b1$ (color magenta).

Algoritmo 7 Ejemplo 2. Filtro de Kalman para estimación de parámetros. Sistema de segundo orden

```

 $n = 2$  //orden del sistema
 $m = 4$  //parámetros del sistema
 $Q = \text{diag}([0,5^20,2^20,5^20,2^2])$  //ruido del proceso
 $R = 0,1$  //ruido de la medición
 $\theta = [0; 0; 0; 0]$  //estimación de parámetros inicial
 $P = 100^m \text{eye}(4)$  //covarianza de estimación de parámetros inicial
 $\theta_{tru} = [1,489; -0,5488; 0,01643; 0,01345]$  //parámetros verdaderos del sistema
 $y1 = 0$  //inicialización salida anterior
 $y2 = 0$  //inicialización salida anterior
 $u1 = 0$  //inicialización entrada
 $u2 = 0$  //inicialización entrada anterior
mediciones o iteraciones
 $N = 50$ 
matrices de almacenamiento
 $yrV = \text{zeros}(1, N)$  //almacenamiento del observador de parámetros (yr)
 $uV = \text{zeros}(1, N)$  //almacenamiento de la señal de control (u)
 $yeV = \text{zeros}(1, N)$  //almacenamiento del observador estimado de parámetros (ye)
 $\theta V = \text{zeros}(m, N)$  //almacenamiento de los parámetros estimados ( $\theta$ )
proceso iterativo
for  $k = 1 : N$  do
  if  $k = 2$ 
     $y1 = yrV(1, k - 1)$  //salida anterior
     $u1 = uV(1, k - 1)$  //entrada anterior
  end if
  if  $k \geq 3$ 
     $y1 = yrV(1, k - 1)$  //salida anterior
     $y2 = yrV(1, k - 2)$  //salida anterior
     $u1 = uV(1, k - 1)$  //entrada anterior
     $u2 = uV(1, k - 2)$  //entrada anterior
  end if
   $u = 10$  //señal de control
   $uV(1, k) = u$  //almacenamiento de la señal de control
   $C = [y1, y2, u1, u2]$  //vector de regresión
   $yr = C\theta_{tru}$  //señal de observación de parámetros
   $yrV(1, k) = yr$  //almacenamiento del observador de parámetros

```

Algoritmo 8 Ejemplo 2. Filtro de Kalman para estimación de parámetros. Sistema de segundo orden, continuación

```
[x,P] =filtro de Kalman-parámetros( $\theta$ ,P,C,yr,Q,R)//realizar una iteración del filtro de Kalman  
 $\theta V(:,k) = \theta$  //almacenamiento de la estimación de parámetros  
 $ye = C\theta$  //señal de observación estimada de parámetros  
 $yeV(1,k) = ye$  //almacenamiento del observador estimado de parámetros  
end for
```

De igual manera si las iteraciones han terminado, las matrices de almacenamiento yrV y yeV contienen todas las observaciones y todas las observaciones estimadas respectivamente, en la figura 2.3 se observa la matriz yrV (color rojo, puntos) y la matriz yeV (color magenta), es posible deducir que las observaciones coinciden con las observaciones estimadas del sistema.

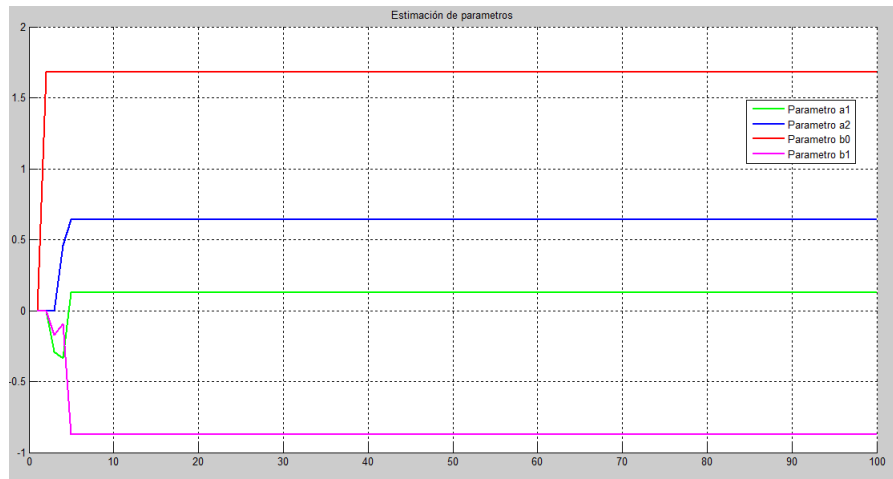


Figura 2.10: Estimación de parámetros mediante el filtro de Kalman. Ejemplo 2 (sistema de segundo orden)

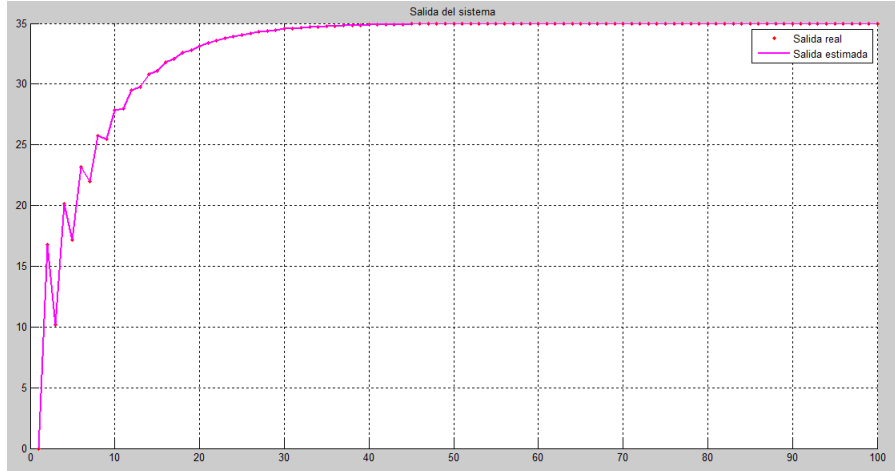


Figura 2.11: Salida estimada mediante el filtro de Kalman. Ejemplo 2 (sistema de segundo orden)

2.3. Filtro de Kalman dual

El filtro de Kalman dual busca estimar los estados y los parámetros de un sistema de manera simultánea, la estimación dual es ampliamente utilizada para diseñar el control de un sistema [19][27].

Una descripción más detallada acerca de estimación dual puede encontrarse en [19] y [27], estas investigaciones se toman como referencia para definir las ecuaciones correspondientes a la estimación dual mediante el filtro de Kalman.

2.3.1. El problema de la estimación dual

Se abordan dos problemáticas al mismo tiempo (problema de la estimación de estados y problema de la estimación de parámetros), las cuales se plantearon anteriormente.

2.3.2. Ecuaciones para estimación dual

- Predicción

- Estimación de parámetros a-priori

$$\underline{\theta}_k = \theta_{k-1} \quad (2.20)$$

- Covarianza de estimación de parámetros a-priori

$$\underline{P}_{\theta_k} = P_{\theta_{k-1}} + Q_{\theta_{k-1}} \quad (2.21)$$

- Estimación de estados a-priori

$$\underline{x}_k = F_{k+1,k} x \quad (2.22)$$

- Covarianza de estimación de estados a-priori

$$\underline{P}_{x_k} = F_{k+1,k} P_{x_{k-1}} F_{k+1,k}^T + Q_{x_{k-1}} \quad (2.23)$$

- Predicción

- Ganancia de Kalman para estimación de estados

$$G_{x_k} = \underline{P}_{x_k} H_k^T [H_k \underline{P}_{x_k} H_k^T + R_{x_k}]^{-1} \quad (2.24)$$

- Estimación de estados a-posteriori

$$x_k = \underline{x}_k + G_{x_k} (zm_k - H_k \underline{x}_k) \quad (2.25)$$

- Covarianza de estimación de estados a-posteriori

$$P_{x_k} = (I - G_{x_k} H_k) \underline{P}_{x_k} \quad (2.26)$$

- Ganancia de Kalman para estimación de parámetros

$$G_{\theta_k} = \underline{P}_{\theta_k} C_k^T [C_k \underline{P}_{\theta_k} C_k^T + R_{\theta_k}]^{-1} \quad (2.27)$$

- Estimación de parámetros a-posteriori

$$\theta_k = \underline{\theta}_k + G_{\theta_k} (y_k - C_k \underline{\theta}_k) \quad (2.28)$$

- Covarianza de estimación de parámetros a-posteriori

$$P_{\theta_k} = (I - G_{\theta_k} C_k) \underline{P}_{\theta_k} \quad (2.29)$$

2.3.3. Algoritmo para estimación dual

Las ecuaciones del filtro de Kalman para la estimación de estados y parámetros se implementan en el algoritmo 9.

Algoritmo 9 Filtro de Kalman dual

Require: $A, x, P_x, H, zm, Q_x, R_x, u, B, \theta, P_\theta, C, y, Q_\theta, R_\theta$

Ensure: x, P_x, θ, P_θ

function $[x, P_x, \theta, P_\theta] = \text{filtro de Kalman dual}(A, x, P_x, H, zm, Q_x, R_x, u, B, \theta, P_\theta, C, y, Q_\theta, R_\theta)$

predicción

$\underline{\theta} = \theta$ // estimación de parámetros a-priori

$\underline{P}_\theta = P_\theta + Q_\theta$ // covarianza de estimación de parámetros a-priori

$\underline{x} = Ax + Bu$ // estimación de estados a-priori

$\underline{P}_x = AP_xA^T + Q_x$ // covarianza de estimación de estados a-priori

corrección

$G_x = \underline{P}_x H^T [H \underline{P}_x H^T + R_x]^{-1}$ // Ganancia de Kalman para estimación de estados

$x = \underline{x} + G_x(zm - H\underline{x})$ // estimación de estados a-posteriori

$P_x = (I - G_x H) \underline{P}_x$ // covarianza de estimación de estados a-posteriori

$G_\theta = \underline{P}_\theta C^T [C \underline{P}_\theta C^T + R_\theta]^{-1}$ // Ganancia de Kalman para estimación de parámetros

$\theta = \underline{\theta} + G_\theta(y - C\underline{\theta})$ // estimación de parámetros a-posteriori

$P_\theta = (I - G_\theta C) \underline{P}_\theta$ // covarianza de estimación de parámetros a-posteriori

return

El algoritmo 9 (filtro de Kalman dual), tiene como objetivo estimar de manera simultánea los estados y los parámetros de un sistema lineal, es posible observar que se requieren las entradas empleadas en el algoritmo para estimación de estados más las entradas empleadas en el algoritmo para estimación de parámetros.

En este caso, el filtro de Kalman dual utiliza de manera paralela los algoritmos contruidos anteriormente para estimar estados y parámetros. La estimación dual también puede ser diseñada de tal forma que a partir de la estimación de parámetros se realice la estimación de estados o viceversa, este enfoque de estimación se conoce como enfoque iterativo.

2.3.4. Ejemplo. Estimación dual

En el ejemplo 2 sobre estimación de estados empleando el filtro de Kalman, se tiene el modelo en espacio de estados de un sistema SISO de segundo orden, a partir del cual se obtienen los parámetros del sistema mediante la herramienta de Matlab “ss2tf”, los cuales se utilizan en el ejemplo 2 sobre estimación de parámetros empleando el filtro de Kalman, por ende si en un “script” (ventana de código) se implementan los dos ejemplos mencionados de manera paralela y se utiliza el filtro de Kalman dual, es posible obtener la misma estimación de estados y parámetros obtenidas de manera individual mediante el filtro de Kalman. En la figura 2.12 las imágenes de la columna izquierda corresponden a la estimación de estados, y las imágenes de la columna derecha corresponden a la estimación de parámetros.

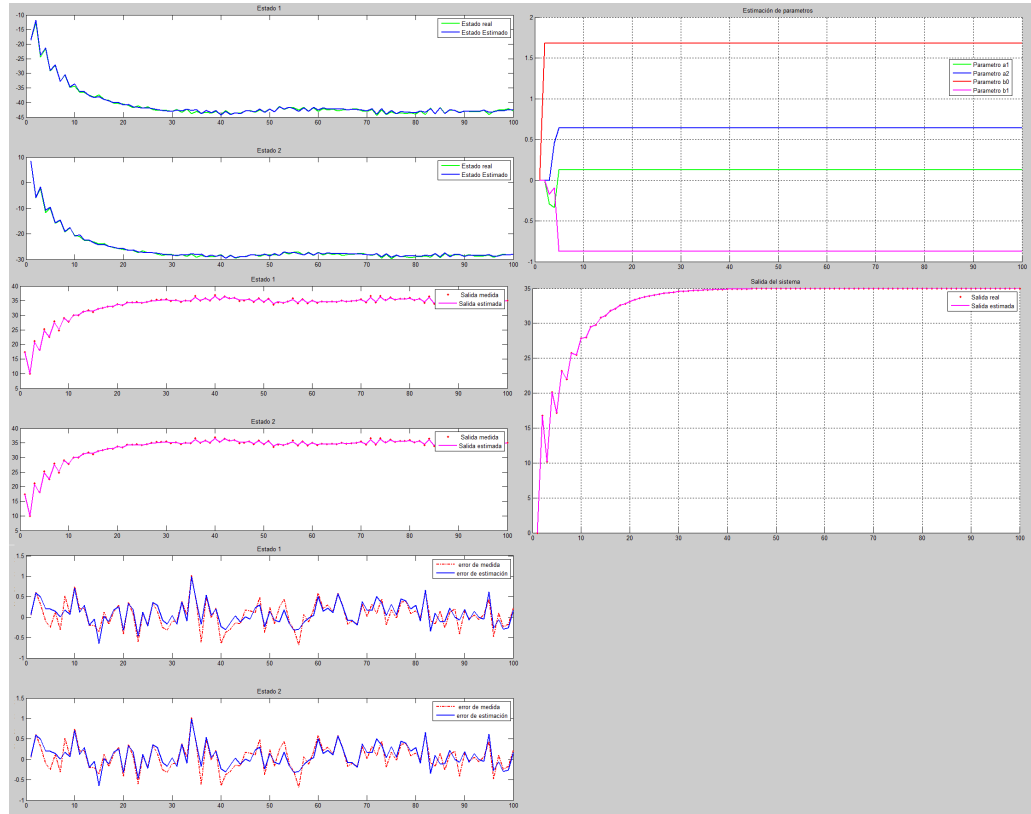


Figura 2.12: Estimación de estados y parámetros mediante el filtro de Kalman dual. Ejemplo (sistema de segundo orden)

3. FILTRO DE KALMAN EXTENDIDO

El filtro de Kalman (KF) es empleado para modelos lineales de sistemas dinámicos, sin embargo cuando se trata de modelos no lineales es posible emplear el filtro de Kalman aplicando un proceso de linealización. El metodo resultante se conoce como filtro de Kalman extendido [19].

Una descripción más detallada acerca del filtro de Kalman extendido puede encontrarse en [19] y [23], estas investigaciones se toman como referencia para definir el problema de la estimación de estados y las ecuaciones del filtro de Kalman extendido.

3.1. Estimación de estados

3.1.1. El problema de la estimación de estados

- Modelo de espacio de estados

Se considera un sistema dinámico no lineal descrito por el siguiente modelo de espacio de estados

$$x_{k+1} = f(k, x_k) + w_k \quad (3.1)$$

Donde la función f denota una matriz de transición de estados no lineal, y x se conoce como vector de estados.

$$y_k = h(k, x_k) + v_k \quad (3.2)$$

Donde la función h denota una matriz de medición no lineal, y y se conoce como vector de observación o salida medida del sistema, además w_k y v_k corresponden al ruido del proceso y al ruido de la medición respectivamente, los ruidos se asumen aditivos, blancos y gaussianos, con media cero y con matriz de covarianza Q_k y R_k respectivamente.

3.1.2. Ecuaciones para estimación de estados

La idea básica del filtro de Kalman extendido es linealizar el modelo de espacio de estados para sistemas lineales en cada instante de tiempo alrededor de la estimación de estado más reciente, una vez se obtiene el modelo lineal es posible aplicar las ecuaciones del filtro de Kalman estándar.

la linealización de la función f y la función h alrededor de la estimación de estado a-priori, con las cuales se obtiene la matriz de transición de estados A y la matriz de medición H se muestran a continuación.

$$A_{k+1,k} = \frac{df(k, x)}{dx} \Big|_{\underline{x}} \quad (3.3)$$

$$H_k = \frac{dh(k, x)}{dx} \Big|_{\underline{x}} \quad (3.4)$$

3.1.3. Algoritmo para estimación de estados

El algoritmo del filtro de Kalman extendido para estimación de estados se muestra en el algoritmo 10 (filtro de Kalman extendido para estimación de estados), el método tiene como objetivo estimar los estados de un sistema no lineal.

Algoritmo 10 Filtro de Kalman extendido para estimación de estados

Require: f, x, P, h, zm, Q, R, n

Ensure: x, P, H

function $[x, P, H] = \text{filtro de Kalman extendido-estados}(f, h, zm, x, P, Q, R, n)$

Predicción

$\underline{x} = f(x) // \text{estimación de estado a-priori}$

$[A, H] = \text{jacobiano}(\underline{x}, f, h, n)$

$\underline{P} = A P A^T + Q // \text{covarianza de estimación a-priori}$

Corrección

$G = \underline{P} H^T [H \underline{P} H^T + R]^{-1} // \text{Ganancia de Kalman}$

$x = \underline{x} + G(z - h(\underline{x})) // \text{estimación de estado a-posteriori}$

$P = (I - GH) \underline{P} // \text{estimación de covarianza a-posteriori}$

return

Se puede observar que el algoritmo necesita como entradas la matriz de transición de estados no lineal f , el estado inicial del sistema x , el orden del sistema n la covarianza de estimación de estados inicial P , la matriz de medición no lineal h , la salida medida del sistema zm o y , y las covarianzas del proceso y medición respectivamente Q, R .

Una vez el algoritmo reconoce las entradas mencionadas anteriormente, se realiza la estimación a-priori de estados, en seguida se realiza un llamado al algoritmo 11 correspondiente al operador jacobiano, el cual entrega la matriz de transición de estados linealizada A y la matriz de medición linealizada H , con la cuales es posible continuar con los pasos descritos anteriormente para el filtro de Kalman estándar (estimación de covarianza a-priori, ganancia de Kalman y finalmente se realiza la estimación a-posteriori de estados y covarianza), las estimaciones a-posteriori son las salidas del algoritmo, además la matriz de medición linealizada H es una salida del algoritmo, ya que es necesaria para calcular la salida estimada del sistema.

Algoritmo 11 Operador jacobiano

Require: \underline{x}, f, h, n

Ensure: A, H

```

function [A,H]=jacobiano( $\underline{x},f,h,n$ )
     $A = \text{zeros}(n,n)$  //matriz de transición de estados inicial
     $H = \text{zeros}(1,n)$  //matriz de medición de estados inicial
     $h_c = \text{neps}/\text{eps} = 2.2204\text{e-}16$ 
    for  $k = 1 : n$  do
         $x1 = \underline{x}$ 
         $x1(k) = x1(k) + h_ci$ 
         $A(:,k) = \text{imag}(f(x1))/h_c$  //aproximadamente  $df/dx$ 
         $H(1,k) = \text{imag}(h(x1))/h_c$  //aproximadamente  $dh/dx$ 
    end for
return
```

El algoritmo 11 (operador jacobiano) tiene como objetivo linealizar las funciones f y h alrededor de la estimación de estado a-priori, de esta manera las estradas del algoritmo son: las funciones f y h las cuales denotan la matriz de transición de estados no lineal y la matriz de medición no lineal, la estimación a-priori de estados \underline{x} y el orden del sistema n .

Una vez el algoritmo reconoce las entradas mencionadas se define la matriz de transición de estados inicial y la matriz de medición inicial, en las cuales se almacenan las derivadas parciales que conforman el jacobiano, luego se define el tiempo de muestreo $hc = neps$ el cual se utiliza para derivar numéricamente las funciones f y h . Una explicación detallada acerca de la derivada numérica y de la obtención del jacobiano puede ser encontrada en la referencia [24] y en la herramienta de MathWorks “complex step jacobian”, estas investigaciones se toman como referencia para definir la derivada numérica de las funciones f y h .

$$f'(\underline{x}) = imag \frac{f(\underline{x} + ihc)}{hc} \quad (3.5)$$

$$h'(\underline{x}) = imag \frac{h(\underline{x} + ihc)}{hc} \quad (3.6)$$

3.1.4. Ejemplo 1. Estimación de estados

A continuación se presenta un ejemplo (algoritmo 12) en el cual se implementara el algoritmo 5 correspondiente al filtro de Kalman extendido. El ejemplo describe un carro-péndulo modelado como un sistema de cuarto orden, la función f es tomada de la referencia [28].

$$f(x) = \begin{bmatrix} x(2) \\ x(2)^2 \sin x(1) \cos x(1) - \sin x(1) \\ x(4) \\ x(2)^2 \sin x(1) \end{bmatrix}$$

Para el modelo del sistema el estado uno corresponde al ángulo entre el péndulo y el eje vertical, el estado dos corresponde a la velocidad angular del péndulo, el estado tres corresponde al desplazamiento del auto y el estado cuatro corresponde a la velocidad del auto. Se puede observar en el algoritmo 12 que en primer lugar se introducen las entradas que requiere el filtro de Kalman extendido para estimación de estados f, x, P, h, Q, R, n con excepción de la salida medida zm , la cual es calculada en el proceso iterativo.

Como paso siguiente se define el número de iteraciones o mediciones N que se realizaran en el algoritmo, luego se definen cinco matrices de almacenamiento: zrV almacenamiento de la salida real del sistema, zmV almacenamiento de mediciones o salida medida (zm), zeV almacenamiento de la salida estimada ($Hx=ze$), xV almacenamiento de la estimación de estados (x), y finalmente $truV$ almacenamiento del estado real del sistema.

Algoritmo 12 Ejemplo 1. Filtro de Kalman Extendido para estimación de estados.
Sistema SISO de cuarto orden

```

n = 4// orden del sistema
f=@(x)[x(2); x(2)x(2)sinx(1)cosx(1)-sinx(1); x(4); x(2)x(2)sinx(1)+sinx(1)cosx(1)]
//matriz de transición de estados no lineal
h = @(x)x(1)//matriz de medición no lineal
Q = diag([0,5^20,2^20,3^20,5^2])//ruido del proceso
R = 0,1//ruido de la medición
x = 1rand(4,1)//estimación de estado inicial
P = 10eye(4)//covarianza de estimación de estado inicial
mediciones o iteraciones
N = 60

matrices de almacenamiento
zrV = zeros(1, N)//almacenamiento de la salida real (zr)
zmV = zeros(1, N)//almacenamiento de mediciones o salida medida (zm)
zeV = zeros(1, N)//almacenamiento de la salida estimada (Hx=ze)
xV = zeros(n, N)//almacenamiento de la estimación de estados (x)
truV = zeros(n, N)//almacenamiento del estado real del sistema (tru)
estado real inicial
tru = 1rand(n, 1)
proceso iterativo
for k = 1 : N do
    tru = f(tru) + sqrt(Q)randn(n, 1)//estado real del sistema
    truV(:, k) = tru //almacenamiento del estado real del sistema
    zm = h(tru) + sqrt(R)randn//medición de los voltajes
    zmV(1, k) = zm//almacenamiento de las mediciones
    zr = h(tru)//salida real del sistema
    zrV(1, k) = zr//almacenamiento de las salidas
    [x, P, H] =filtro de Kalman extendido-estados(f, h, zm, x, P, Q, R, n)//realizar una
    iteración del filtro de Kalman extendido
    xV(:, k) = x //almacenamiento de la estimación de estados
    zeV(:, k) = Hx//almacenamiento de la salida o mediciones estimadas
end for
cálculo de errores
MeasErr = zr - zmV//error de medición
EstErr = zr - zeV//error de estimación
MeasErrCov = sum(MeasErrMeasErr)/length(MeasErr)//covarianza del error
de medición
EstErrCov = sum(EstErrEstErr)/length(EstErr)//covarianza del error de esti-
mación

```

Cabe mencionar que un sistema es no lineal si las ecuaciones que regulan su comportamiento son no lineales, es decir cuando estas ecuaciones no son de primer grado, una ecuación de primer grado se caracteriza por no tener productos entre sus variables [25]. Se puede observar que la función f es de carácter no lineal, la función h permite elegir qué estado será estimado, de esta manera se elige h como la primera posición del vector de estados $x(1)$, lo cual representa que la estimación será realizada de manera óptima únicamente para el primer estado del sistema, para los estados restantes la estimación a-posteriori será igual a la a-priori $x = \underline{x}$, ya que solo se tiene ganancia de Kalman G en el primer estado.

Posteriormente se define un valor inicial del estado real del sistema tru , luego se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N que este indicado, en el proceso iterativo se calcula el estado real del sistema tru , el cual se describe de acuerdo con la ecuación 3.1, tru es almacenado en la matriz $truV$, a continuación se realiza la medición del sistema zm , la cual se describe mediante la ecuación 3.2 y se almacena en la matriz zmV , después se calcula la salida real del sistema zr , la cual es igual a la salida medida sin tener en cuenta el ruido de la medición, esta salida se almacena en zrV .

Como paso siguiente se realiza una iteración del filtro de Kalman extendido para estimación de estados, por ende como se explicó anteriormente se obtiene la estimación de estados x , y la covarianza de estimación de estado P , las estimaciones mencionadas se utilizan en la siguiente iteración como estimaciones iniciales, x es almacenada en su respectiva matriz xV , además la matriz de medición linealizada H es una salida del filtro de Kalman extendido para estimación de estados, con la cual es posible obtener la salida estimada del sistema ze , es decir el producto entre H y la estimación de estados x , esta salida es almacenada en zeV , por último se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y las covarianzas de error $MeasErrCov$ y $EstErrCov$.

Una vez las iteraciones han terminado, las matrices de almacenamiento $truV$ y xV contienen todos los estados verdaderos y todos los estados estimados respectivamente, como se explicó anteriormente la estimación de estado se realiza únicamente para el primer estado del sistema, de esta manera se puede observar en la figura 3.1 la matriz $truV$ (color verde) y la matriz xV (color azul), es posible deducir que el estado estimado se aproxi-

ma de manera considerable al estado verdadero respecto al primer estado del sistema, además se observa que la estimación de estados únicamente se realiza para el primer estado del sistema.

De igual manera, si las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas verdaderas y todas las salidas estimadas respectivamente, las matrices zmV y zeV son matrices de tamaño $(1 \times N)$ debido a que se tiene una sola salida en el sistema, por ende las salidas reales y estimadas son las mismas para los cuatro estados del sistema, se puede observar en la figura 3.2 la matriz zmV (color rojo, puntos) y zeV (color magenta), es posible observar que las salidas estimadas se aproximan a las salidas medidas.

Finalmente en la figura 2.10 se puede observar la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y entre el error de estimación $EstErr$ (color azul), la figura muestra que el nivel de ruido es reducido una vez se ha implementado el filtro de Kalman, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, para el sistema de interés la covarianza de error antes de la implementación del filtro ($MeasErrCov$) es igual a 0,0972 y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 0,0777.

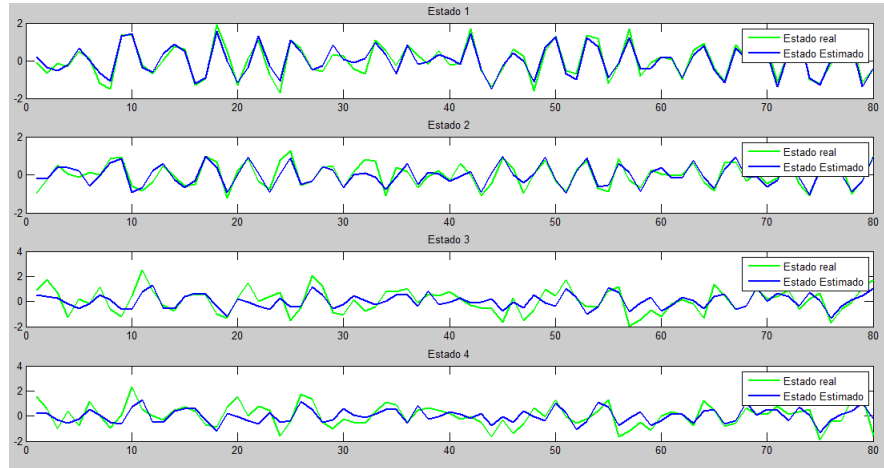


Figura 3.1: Estimación de estados mediante el filtro de Kalman Extendido. Ejemplo 1 (sistema de cuarto orden)

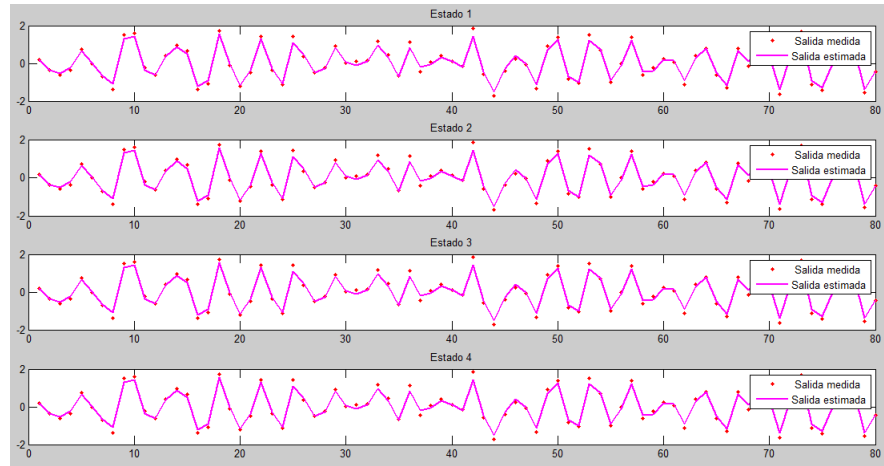


Figura 3.2: Salidas estimadas mediante el filtro de Kalman Extendido. Ejemplo 1 (sistema de cuarto orden)

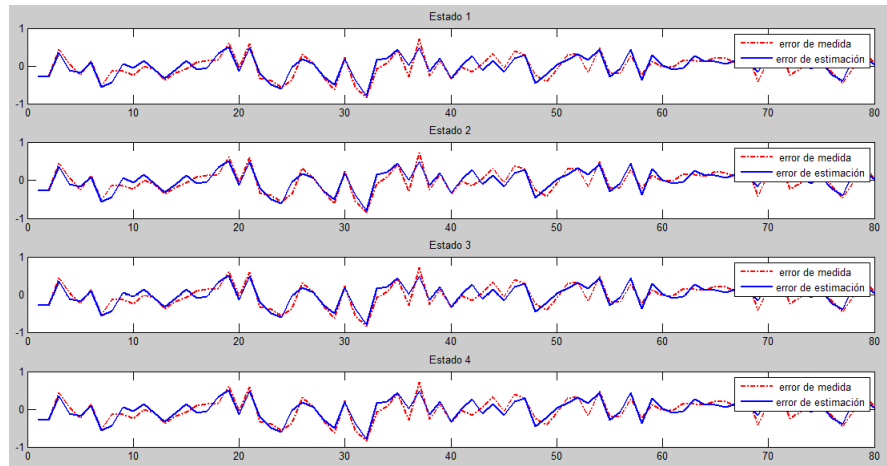


Figura 3.3: Error de estimación mediante el filtro de Kalman Extendido. Ejemplo 1 (sistema de cuarto orden)

3.1.5. Ejemplo 2. Estimación de estados

A continuación se presenta un ejemplo (algoritmo 13) en el cual se implementa el algoritmo 10 correspondiente al filtro de Kalman extendido para estimación de estados. El ejemplo describe un sistema SISO (una entrada, una salida) de segundo orden y de comportamiento no lineal, la función f del ejemplo es tomada de la herramienta de Math-Works “learning ensemble Kalman filter”.

$$f(x) = \begin{bmatrix} x(1) + 0,1x(2) + 0,005 \\ x(2) + 0,1 \end{bmatrix}$$

En el algoritmo 13 se introducen las entradas que requiere el filtro de Kalman extendido para estimación de estados con excepción de la salida medida zm , la cual es calculada en el proceso iterativo, como paso siguiente se define el número de iteraciones N que serán realizadas, luego se definen cinco matrices de almacenamiento, la función h permite elegir cual estado del sistema será estimado, se elige h como la primera posición del vector de estados $x(1)$, posteriormente se define un valor inicial del estado real del sistema tru ,

Se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N que este indicado, se calcula el estado real del sistema tru , la medición del sistema zm , y la salida real del sistema zr , las cuales son almacenadas en sus respectivas matrices. Como paso siguiente se realiza una iteración del filtro de Kalman extendido para estimación de estados, por ende se obtiene la estimación de estados x , y la covarianza de estimación de estado P , x es almacenada en su respectiva matriz xV , además se calcula la salida estimada del sistema ze , esta salida es almacenada en zeV , luego se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y las covarianzas de error $MeasErrCov$ y $EstErrCov$.

En la figura 3.4 se observa la comparación de las matrices $truV$ (color verde) y xV (color azul), las cuales contienen todos los estados verdaderos y todos los estados estimados respectivamente, la estimación de estado se realiza únicamente para el primer estado del sistema, en la figura 3.5 se observa la comparación de las matrices zmV (color rojo, puntos) y zeV (color magenta), las cuales contienen todas las salidas medidas y todas las salidas estimadas respectivamente.

Algoritmo 13 Ejemplo 2. Filtro de Kalman Extendido para estimación de estados. Sistema SISO de segundo orden

```

n = 2// orden del sistema
f=@(x)[x(1) + 0,1x(2) + 0,005; x(2) + 0,1]//matriz de transición de estados no lineal
h = @(x)x(1)//matriz de medición no lineal
Q = Q = 0,1 * eye(n)//ruido del proceso
R = 1//ruido de la medición
x = 0,1rand(2,1)//estimación de estado inicial
P = 10eye(2)//covarianza de estimación de estado inicial
mediciones o iteraciones
N = 80

matrices de almacenamiento
zrV = zeros(1, N)//almacenamiento de la salida real (zr)
zmV = zeros(1, N)//almacenamiento de mediciones o salida medida (zm)
zeV = zeros(1, N)//almacenamiento de la salida estimada (Hx=ze)
xV = zeros(n, N)//almacenamiento de la estimación de estados (x)
truV = zeros(n, N)//almacenamiento del estado real del sistema (tru)
estado real inicial
tru = [1; 1]
proceso iterativo
for k = 1 : N do
    tru = f(tru) + sqrt(Q)randn(n, 1)//estado real del sistema
    truV(:, k) = tru //almacenamiento del estado real del sistema
    zm = h(tru) + sqrt(R)randn//medición de los voltajes
    zmV(1, k) = zm//almacenamiento de las mediciones
    zr = h(tru)//salida real del sistema
    zrV(1, k) = zr//almacenamiento de las salidas reales del sistema
    [x, P, H] =filtro de Kalman extendido-estados(f, h, zm, x, P, Q, R, n)//realizar una
    iteración del filtro de Kalman extendido
    xV(:, k) = x //almacenamiento de la estimación de estados
    zeV(:, k) = Hx//almacenamiento de la salida o mediciones estimadas
end for
cálculo de errores
MeasErr = zr - zmV//error de medición
EstErr = zr - zeV//error de estimación
MeasErrCov = sum(MeasErrMeasErr)/length(MeasErr)//covarianza del error
de medición
EstErrCov = sum(EstErrEstErr)/length(EstErr)//covarianza del error de esti-
mación

```

Finalmente en la figura 3.6 se puede observar la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y entre el error de estimación $EstErr$ (color azul), la figura muestra que el nivel de ruido es reducido una vez se ha implementado el filtro de Kalman, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, la covarianza de error antes de la implementación del filtro ($MeasErrCov$) es igual a 0,9240 y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 0,3545.

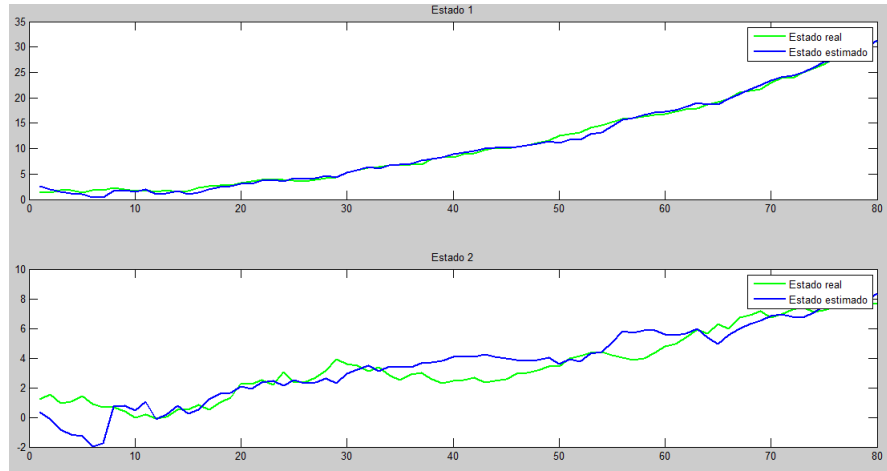


Figura 3.4: Estimación de estados mediante el filtro de Kalman Extendido. Ejemplo 2 (sistema de segundo orden)

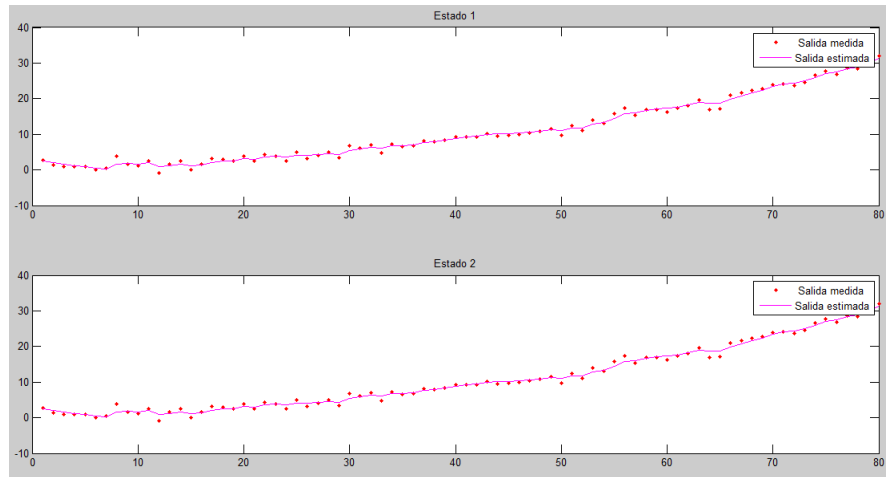


Figura 3.5: Salidas estimadas mediante el filtro de Kalman Extendido. Ejemplo 2 (sistema de segundo orden)

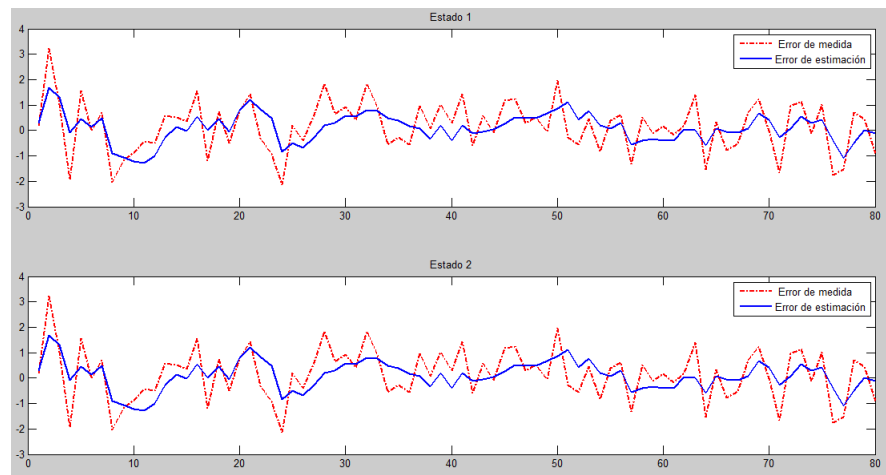


Figura 3.6: Error de estimación mediante el filtro de Kalman Extendido. Ejemplo 2 (sistema de segundo orden)

3.2. Estimación de parámetros

El filtro de Kalman extendido también puede ser empleado para estimar los parámetros de un sistema no lineal. La estimación de parámetros también es conocida como identificación de sistemas o aprendizaje de maquina [19].

Una descripción más detallada acerca de la estimación de parámetros a partir del filtro de Kalman extendido puede encontrarse en [19], esta investigación se toma como referencia para definir el problema y las ecuaciones correspondientes a la estimación de parámetros mediante el filtro de Kalman extendido.

3.2.1. El problema de la estimación de parámetros

Para estimar los parámetros de un sistema no lineal, se considera la siguiente representación en espacio de estados.

$$\theta_k = \theta_{k-1} + w_k \quad (3.7)$$

$$y_k = G(x_k, \theta_k) + v_k \quad (3.8)$$

Donde y_k se conoce como el observador de parámetros, G corresponde a una función parametrizada no lineal, θ_k corresponde a un proceso estacionario con matriz de transición de estados igual a la matriz identidad, además para procesos estacionarios es posible despreocuparse los ruidos de proceso y medición w_k y v_k en la representación en espacio de estados.

3.2.2. Ecuaciones para estimación de parámetros

La idea básica del filtro de Kalman extendido para estimación de parámetros es linealizar la función parametrizada no lineal G para cada instante de tiempo alrededor de la estimación de parámetros más reciente, una vez se obtiene la función linealizada es posible aplicar las ecuaciones del filtro de Kalman estándar para estimación de parámetros.

La linealización de la función G alrededor de la estimación de parámetros a-priori, con la cual se obtiene el vector C se muestran a continuación.

$$C_k = \frac{dG(x_k, \theta_k)}{d\theta} \Big|_{\underline{\theta}_k} \quad (3.9)$$

3.2.3. Algoritmo para estimación de parámetros

El algoritmo del filtro de Kalman extendido para estimación de parámetros se muestra en el algoritmo 14 (filtro de Kalman extendido para estimación de parámetros).

Algoritmo 14 Filtro de Kalman extendido para estimación de parámetros

Require: G, θ, P, y, Q, R, m

Ensure: θ, P

function $[\theta, P] = \text{filtro de Kalman extendido-parámetros}((G, \theta, P, y, Q, R, m))$

Predicción

$\underline{\theta} = \theta // \text{estimación de parámetros a-priori}$

$\underline{P} = P + Q // \text{covarianza de estimación a-priori}$

$[C] = \text{jacobiano}(\underline{\theta}, G, m)$

Corrección

$G = \underline{P}C^T[C\underline{P}C^T + R]^{-1} // \text{ganancia de Kalman}$

$\theta = \underline{\theta} + G(y - G(\underline{\theta})) // \text{estimación de parámetros a-posteriori}$

$P = (I - GC)\underline{P} // \text{covarianza de estimación a-posteriori}$

return

El método tiene como objetivo estimar los parámetros de un sistema no lineal, se puede observar que el algoritmo necesita como entradas la función parametrizada no lineal G , la estimación de parámetros inicial θ , el número de parámetros del sistema m la covarianza de estimación de parámetros inicial P , el observador de parámetros y , y las covarianzas del proceso y medición respectivamente Q, R .

Una vez el algoritmo reconoce las entradas mencionadas anteriormente, se realiza la estimación a-priori de parámetros y de covarianza (predicción), en seguida se realiza un llamado al algoritmo 15 correspondiente al operador jacobiano, el cual entrega la linealización de la función parametrizada no lineal C (matriz de medición de parámetros), con la cual es posible continuar con los pasos descritos anteriormente para el filtro de Kalman para estimación de parámetros, se calcula la ganancia de Kalman y finalmente se

realiza la estimación a-posteriori de parámetros y covarianza (corrección), las estimaciones a-posteriori son las salidas del algoritmo. Se considera al filtro de Kalman extendido como un algoritmo de solución recursiva debido a que los parámetros estimados θ y la covarianza de estimación P calculados en una iteración son empleados en la próxima iteración como estimación de parámetros inicial y covarianza de estimación inicial.

Algoritmo 15 Operador jacobiano

Require: $\underline{\theta}, G, m$

Ensure: C

```

function [C]=jacobiano( $\underline{\theta}, G, m$ )
 $C = \text{zeros}(1, m)$  //matriz de medición de parámetros inicial
 $h_c = \text{meps} // \text{eps} = 2.2204\text{e-}16$ 
for  $k = 1 : m$  do
     $\theta_1 = \underline{\theta}$ 
     $\theta_1(k) = \theta_1(k) + h_c i$ 
     $C(1, k) = \text{imag}(h(\theta_1)) / h_c$  //aproximadamente  $dG/d\theta$ 
end for
return

```

El algoritmo 15 (operador jacobiano) tiene como objetivo linealizar la función G alrededor de la estimación de parámetros a-priori, las estradas del algoritmo son: la función parametrizada no lineal G , la estimación a-priori de parámetros $\underline{\theta}$ y el número de parámetros del sistema m .

Una vez el algoritmo reconoce las entradas mencionadas se define la matriz de medición de parámetros inicial, en la cual se almacenan las derivadas parciales que conforman el jacobiano, luego se define el tiempo de muestreo $h_c = \text{meps}$ la cual se utiliza para derivar numéricamente la función G .

Una explicación detallada acerca de la derivada numérica y de la obtención del jacobiano puede ser encontrada en la referencia [24] y en la herramienta de MathWorks “complex step jacobian”, estas investigaciones se toman como referencia para definir la derivada numérica de la función G .

$$G'(\underline{\theta}) = \text{imag} \frac{G(\underline{\theta} + i h_c)}{h_c} \quad (3.10)$$

3.2.4. Ejemplo. Estimación de parámetros

Se presenta un ejemplo (algoritmo 16) en el cual se implementa el algoritmo 14 correspondiente al filtro de Kalman extendido para estimación de parámetros. En el ejemplo se presenta la función Rosenbrock [19].

$$f(x_1 + x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (3.11)$$

Se desea encontrar los valores de x_1 y x_2 que minimicen la función, de esta manera los valores óptimos de los parámetros son $x_1 = 1$ y $x_2 = 1$, la función Rosenbrock es empleada para comparar tasas de convergencia en diversos métodos de optimización, la función consta de dos parámetros [19].

Se puede observar en el algoritmo 16 que en primer lugar se introducen las entradas que requiere el filtro de Kalman extendido para estimación de parámetros (θ, P, G, Q, R, m) con excepción de la señal de observación de parámetros y , la cual se calculan en el proceso iterativo. Como paso siguiente se define el número de iteraciones o mediciones N que se realizan en el algoritmo, luego se definen tres matrices de almacenamiento: yV almacenamiento del observador de parámetros (y), yeV almacenamiento del observador estimado de parámetros (ye), y finalmente θV almacenamiento de la estimación de parámetros (θ).

A continuación se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N que este indicado, luego se define el observador de parámetros verdaderos y (ecuación 3.8), el cual se almacena en la matriz yrV , posteriormente se realiza una iteración del filtro de Kalman extendido para estimación de parámetros, por ende como se explicó anteriormente se obtiene la estimación de parámetros θ , la covarianza de estimación P , y la matriz de medición de parámetros C , las estimaciones se utilizan como estimaciones iniciales en la siguiente iteración, θ se almacena en la matriz θV , se prosigue calculando la señal de observación estimada de parámetros ye multiplicando la matriz de medición de parámetros C por los parámetros estimados θ , finalmente ye es almacenado en yeV .

Algoritmo 16 Ejemplo. Filtro de Kalman extendido para estimación de parámetros. Función Rosenbrock

```

m = 2//parámetros del sistema
G = @(θ)[100((θ(2) - (θ(1)2))2)+(1 - θ(1))2]//funcion Rosenbrock
Q = 0,1eye(m) //ruido del proceso
R = 0,1 //ruido de la medición
θ = [0,2; 0,3]//estimación de parámetros inicial
P = 100meye(m)//covarianza de estimación de parámetros inicial
θtru = [1; 1]//parámetros reales
mediciones o iteraciones
N = 500
matrices de almacenamiento
yV = zeros(1, N)//almacenamiento del observador de parámetros (y)
yeV = zeros(1, N)//almacenamiento del observador estimado de parámetros (ye)
θV = zeros(m, N)//almacenamiento de los parámetros estimados (θ)
proceso iterativo
for k = 1 : N do
    y = G(θtru)//señal de observación de parámetros
    yV(1, k) = y//almacenamiento del observador de parámetros
    [θ, P, C] = filtro de Kalman extendido-parámetros((G, θ, P, y, Q, R, m)//realizar
    una iteración del filtro de Kalman extendido.
    θV(:, k) = θ //almacenamiento de la estimación de parámetros
    ye = Cθ//señal de observación estimada de parámetros
    yeV(1, k) = ye//almacenamiento del observador estimado de parámetros
end for

```

Es importante mencionar que el algoritmo realiza la estimación de parámetros utilizando cada medición (observador de parámetros y) tan pronto como esté disponible, por ende se considera que el algoritmo realiza una estimación on-line, además la covarianza Q representa el ruido del proceso, para el caso de estimación de estados, el proceso hace referencia a la evolución de estados (ecuación 3.1), para el caso de estimación de parámetros, el proceso tiene naturaleza estacionaria (ecuación 3.7); Por otro lado, la covarianza R representa el ruido de medición, para el caso de estimación de estados, medición hace referencia a la salida medida del sistema (ecuación 3.2)), para el caso de estimación de parámetros, medición hace referencia al observador de parámetros del sistema (ecuación 3.8).

Una vez las iteraciones han terminado, la matriz de almacenamiento θV contienen todos los parámetros estimados, en la figura 3.7 es posible observar que para la iteración quinientos los parámetros estimados se aproximan de manera considerable a los parámetros reales, como se mencionó anteriormente, en este caso se estiman dos parámetros, parámetro 1 (color verde) y parámetro 2 (color azul).

De igual manera si las iteraciones han terminado, las matrices de almacenamiento yV y yeV contienen todas las observaciones y todas las observaciones estimadas respectivamente, es importante destacar que las matrices mencionadas son de tamaño $1 \times N$, debido a que el sistema tiene una sola salida, en la figura 3.8 se observa la matriz yV (color rojo, puntos) y la matriz yeV (color magenta), es posible deducir que las observaciones se aproximan a las observaciones estimadas del sistema.

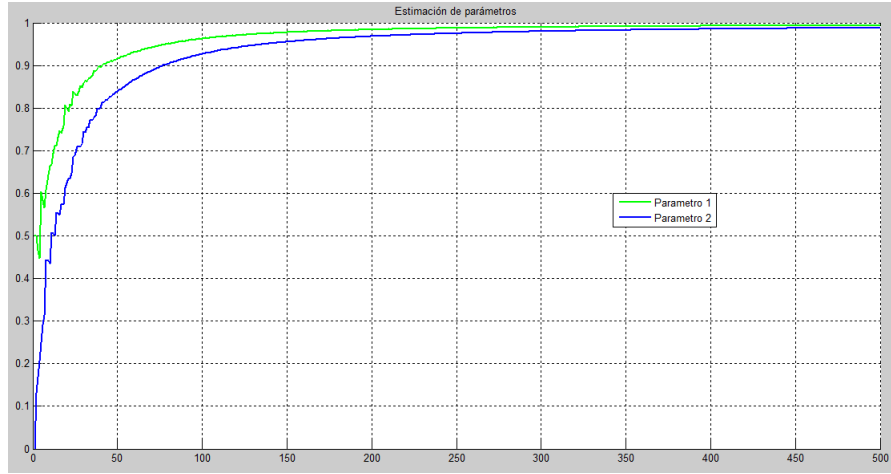


Figura 3.7: Estimación de parámetros mediante el filtro de Kalman extendido. Ejemplo (función Rosenbrock)

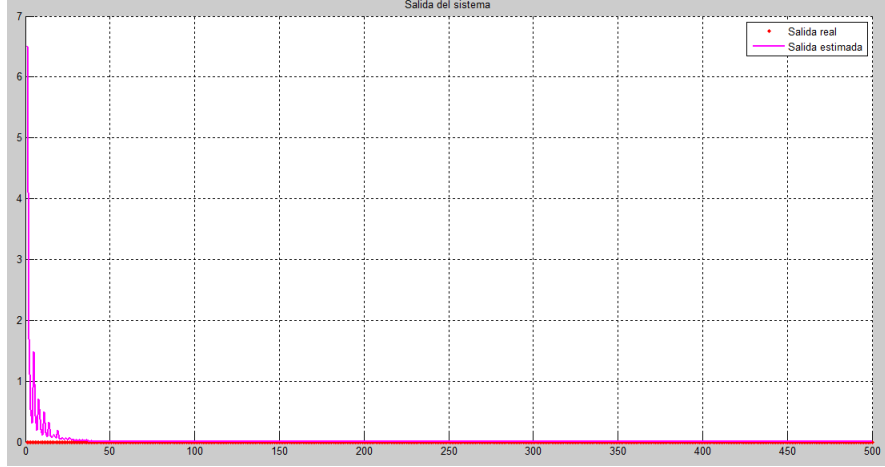


Figura 3.8: Salida estimada mediante el filtro de Kalman extendido. Ejemplo (función Rosenbrock)

3.3. Filtro de Kalman extendido dual

El filtro de Kalman extendido dual busca estimar los estados y los parámetros de un sistema no lineal de manera simultánea, a partir de un solo observador. El filtro de Kalman extendido dual, consta de dos algoritmos, los cuales se ejecutan de manera simultánea, un algoritmo estima los estados del sistema empleando la estimación actual de parámetros, mientras tanto, el otro algoritmo estima los parámetros del sistema empleando la estimación actual de estados. El funcionamiento del algoritmo dual se presenta en la figura 3.11 [19].

Es posible observar que los parámetros del sistema θ son empleados en la predicción de la estimación de estados, debido a que son evaluados en la función f , además los estados y parámetros a-priori son necesarios para el observador y , debido a que f está en términos de x y θ . Se considera que la estimación tiene un enfoque secuencial, debido a que se utiliza una observación a la vez [19].

Una descripción más detallada acerca del filtro de Kalman extendido dual puede encontrarse en [19], esta investigación se toma como referencia para definir el problema de estimación, y las ecuaciones correspondientes a la estimación dual.

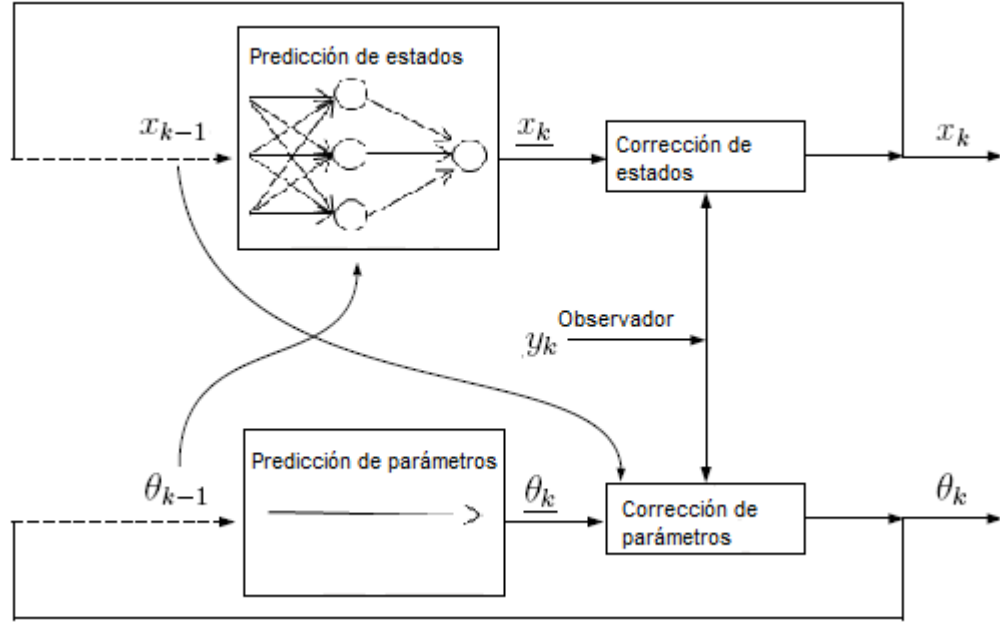


Figura 3.9: Estimación de estados y parámetros mediante el filtro de Kalman extendido dual. Esquema de funcionamiento

3.3.1. El problema de la estimación dual

Se considera el siguiente modelo en espacio de estados:

$$x_{k+1} = f(x_k, u_k, \theta) + v_k \quad (3.12)$$

$$y_k = h(x_k) + n_k \quad (3.13)$$

w_k y v_k corresponden al ruido del proceso y al ruido de la medición respectivamente, los ruidos se asumen aditivos, blancos y gaussianos, con media cero y con matriz de covarianza Q_k y R_k respectivamente.

3.3.2. Ecuaciones para estimación dual

- Predicción

- Estimación de parámetros a-priori

$$\underline{\theta}_k = \theta_{k-1} \quad (3.14)$$

- Covarianza de estimación de parámetros a-priori

$$\underline{P}_{\theta_k} = P_{\theta_{k-1}} + Q_{\theta_{k-1}} \quad (3.15)$$

- Estimación de estados a-priori

$$\underline{x}_k = f(x_{k-1}, u_k, \underline{\theta}_k) \quad (3.16)$$

- Covarianza de estimación de estados a-priori

$$\underline{P}_{x_k} = F_{k+1,k} P_{x_{k-1}} F_{k+1,k}^T + Q_{x_{k-1}} \quad (3.17)$$

- Predicción

- Ganancia de Kalman para estimación de estados

$$G_{x_k} = \underline{P}_{x_k} C^T [C \underline{P}_{x_k} C^T + R_{x_k}]^{-1} \quad (3.18)$$

- Estimación de estados a-posteriori

$$x_k = \underline{x}_k + G_{x_k} (y_k - h(\underline{x}_k)) \quad (3.19)$$

- Covarianza de estimación de estados a-posteriori

$$P_{x_k} = (I - G_{x_k} C) \underline{P}_{x_k} \quad (3.20)$$

- Ganancia de Kalman para estimación de parámetros

$$G_{\theta_k} = \underline{P}_{\theta_k} C_{\theta_k}^T [C_{\theta_k} \underline{P}_{\theta_k} C_{\theta_k}^T + R_{\theta_k}]^{-1} \quad (3.21)$$

- Estimación de parámetros a-posteriori

$$\theta_k = \underline{\theta}_k + G_{\theta_k} (y_k - C_{\theta_k} \underline{\theta}_k) \quad (3.22)$$

- Covarianza de estimación de parámetros a-posteriori

$$P_{\theta_k} = (I - G_{\theta_k} C_{\theta_k}) \underline{P_{\theta_k}} \quad (3.23)$$

Donde se realizan las linealizaciones.

$$F_{k-1} = \left. \frac{df(x_{k-1}, u_k, \underline{\theta_k})}{dx} \right|_{\underline{x}} \quad (3.24)$$

$$C = \left. \frac{dh(x_k)}{dx} \right|_{\underline{x}} \quad (3.25)$$

$$C_{\theta_k} = C \left. \frac{d\underline{x}}{d\underline{\theta}} \right|_{\underline{\theta}} \quad (3.26)$$

3.3.3. Algoritmo para estimación dual

Las ecuaciones del filtro de Kalman extendido dual se implementan en el algoritmo 17 (filtro de Kalman extendido dual), tiene como objetivo estimar de manera simultánea los estados y los parámetros de un sistema no lineal, es posible observar que se requieren las entradas empleadas en el algoritmo para estimación de estados más las entradas empleadas en el algoritmo para estimación de parámetros.

En primer lugar se realiza la estimación a-priori de parámetros y covarianza, luego se realiza la estimación de estados a-priori, con la cual es posible realizar un llamado al operador jacobiano (algoritmo 18), el cual tiene como objetivo linealizar las funciones f y h alrededor de la estimación es estado a-priori, de esta manera las estradas del algoritmo son: las funciones f y h las cuales denotan la matriz de transición de estados no lineal y la matriz de medición de estados no lineal, la estimación a-priori de estados \underline{x} y el orden del sistema n .

Una vez el algoritmo 18 reconoce las entradas mencionadas se define la matriz de transición de estados inicial y la matriz de medición inicial, en las cuales serán almacenadas las derivadas parciales que conforman el jacobiano, luego se define el tiempo de muestreo $hc = neps$ el cual se utiliza para derivar numéricamente las funciones f y h . A continuación se define la matriz de medición de parámetros, multiplicando H por la función G la cual se obtiene derivando la función f respecto a los parámetros, cabe mencionar los estados estimados a-priori, deben ser evaluados en G ; para terminar el paso de predicción se calcula la covarianza de estimación de estados a-priori.

Algoritmo 17 Filtro de Kalman extendido dual

Require: $f, h, x, P_x, y, Q_x, R_x, \theta, P_\theta, Q_\theta, R_\theta, n, G$ **Ensure:** $x, P_x, \theta, P_\theta, H$

function $[x, P_x, \theta, P_\theta, H]$ =filtro de Kalman extendido dual($f, h, x, P_x, y, Q_x, R_x, \theta, P_\theta, Q_\theta, R_\theta, n, G$)
predicción
 $\underline{\theta} = \theta$ // estimación de parámetros a-priori
 $\underline{P}_\theta = P_\theta + Q_\theta$ // covarianza de estimación de parámetros a-priori
 $\underline{x} = f(x)$ // estimación de estados a-priori
 $[A, H]$ =jacobiano(\underline{x}, f, h, n)
 $C = HG(\underline{x})$ // matriz de medición de parámetros
 $\underline{P}_x = AP_xA^T + Q_x$ // covarianza de estimación de estados a-priori
corrección
 $G_x = \underline{P}_xH^T[H\underline{P}_xH^T + R_x]^{-1}$ // Ganancia de Kalman para estimación de estados
 $x = \underline{x} + G_x(y - H\underline{x})$ // estimación de estados a-posteriori
 $P_x = (I - G_xH)\underline{P}_x$ // covarianza de estimación de estados a-posteriori
 $G_\theta = \underline{P}_\theta C^T[C\underline{P}_\theta C^T + R_\theta]^{-1}$ // Ganancia de Kalman para estimación de parámetros
 $\theta = \underline{\theta} + G_\theta(y - C\underline{\theta})$ // estimación de parámetros a-posteriori
 $P_\theta = (I - G_\theta C)\underline{P}_\theta$ // covarianza de estimación de parámetros a-posteriori
return

Algoritmo 18 Operador jacobiano

Require: \underline{x}, f, h, n **Ensure:** A, H

function $[A, H]$ =jacobiano(\underline{x}, f, h, n)
 $A = \text{zeros}(n, n)$ // matriz de transición de estados inicial
 $H = \text{zeros}(1, n)$ // matriz de medición de estados inicial
 $h_c = \text{neps} / \text{eps} = 2.2204\text{e-}16$
for $k = 1 : n$ **do**
 $x1 = \underline{x}$
 $x1(k) = x1(k) + h_c i$
 $A(:, k) = \text{imag}(f(x1)) / h_c$ // aproximadamente df/dx
 $H(1, k) = \text{imag}(h(x1)) / h_c$ // aproximadamente dh/dx
end for
return

Una vez se ha realizado la predicción de estados y parámetros se da lugar a la corrección de los mismos, se calcula la ganancia de Kalman para estados, con la cual es posible obtener la estimación de estados a-posteriori y la covarianza de estimación de estados a-posteriori, de la misma forma se calcula la ganancia de Kalman para parámetros, con la cual es posible obtener la estimación de parámetros a-posteriori y la covarianza de estimación de parámetros a-posteriori. Es posible observar en el algoritmo 17, que para el cálculo de las ganancias de estados y parámetros se utiliza el mismo observador.

3.3.4. Ejemplo. Estimación dual

A continuación se presenta un ejemplo (algoritmo 19) en el cual se implementa el algoritmo 17 correspondiente al filtro de Kalman extendido dual. El ejemplo describe una maquina síncrona modelada como un sistema de cuarto orden, la función f es tomada de la referencia [29].

$$f(x) = \begin{bmatrix} x(1) - 10(\theta x(1) - 3,1416x(3)\sin(x(4)) - 2)/30 \\ x(2) - 10(\theta x(2) + 3,1416x(3)\cos(x(4)) - 2)/30 \\ x(3) \\ x(4) + 10x(3) \end{bmatrix}$$

Para el modelo del sistema el estado uno y el estado dos corresponden a corrientes del sistema, el estado tres corresponde a la frecuencia de operación y el estado cuatro corresponde al desfase de la máquina; el parámetro del sistema corresponde a una resistencia. Se puede observar en el algoritmo 19,20 que en primer lugar se introducen las entradas que requiere el filtro de Kalman extendido dual $f, h, x, P_x, y, Q_x, R_x, \theta, P_\theta, Q_\theta, R_\theta, n, G$ con excepción del observador y , el cual se calcula en el proceso iterativo.

Se define el número de parámetros m , el estado real del sistema tru , y el número de iteraciones o mediciones N que se realizan en el algoritmo, luego se definen seis matrices de almacenamiento: yV almacenamiento del observador del sistema, zrV almacenamiento de la salida real, zeV almacenamiento de la salida estimada, xV almacenamiento de la estimación de estados, $truV$ almacenamiento del estado real del sistema y finalmente θV almacenamiento de los parámetros estimados.

Algoritmo 19 Ejemplo. Filtro de Kalman Extendido dual. Maquina síncrona

```
n = 4// orden del sistema
m = 1// parámetros del sistema
f = @(x)[x(1) - 10(thetax(1) - 3,1416x(3)sin(x(4)) - 2)/30; x(2) - 10(thetax(2) +
3,1416x(3)cos(x(4)) - 2)/30; x(3); x(4) + 10x(3)] //matriz de transición de estados
no lineal
G = @(x)[x(1)/3; x(2)/3; 0; 0] // derivada de f respecto a θ
h = @(x)x(1)//salida del sistema.
datos para estimación de estados
Qx = 0,1eye(n)//ruido del proceso
Rx = 0,1//ruido de la medición
x = ones(n, 1)//estimación de estado inicial
Px = 10eye(n)//covarianza de estimación de estados inicial
tru = [1; 1; 1; 1]//estado real del sistema
datos para estimación de parámetros
Qθ = 0,1eye(m)//ruido del proceso
Rθ = 0,1//ruido de la medición
θ = 0,6//estimación de parámetros inicial
Pθ = 1000eye(m)//covarianza de estimación de parámetros inicial
mediciones o iteraciones
N = 100
matrices de almacenamiento
zrV = zeros(1, N)//almacenamiento de la salida real (zr)
yV = zeros(1, N)//almacenamiento del observador (y)
zeV = zeros(1, N)//almacenamiento de la salida estimada (Hx=ze)
xV = zeros(n, N)//almacenamiento de la estimación de estados (x)
truV = zeros(n, N)//almacenamiento del estado real del sistema (tru)
θV = zeros(m, N)//almacenamiento de la estimación de parámetros θ
proceso iterativo
for k = 1 : N
tru = f(tru) + sqrt(Qx)randn(n, 1)//estado real del sistema
truV(:, k) = tru //almacenamiento del estado real del sistema
y = h(tru) + sqrt(Rx)randn//observador del sistema.
yV(1, k) = y//almacenamiento del observador
zr = h(tru)//salida real del sistema
zrV(1, k) = zr//almacenamiento de las salidas
[x, Px, θ, Pθ, H]=filtro de Kalman extendido dual(f, h, x, Px, y, Qx, Rx, θ, Pθ,
Qθ, Rθ, n, G//realizar una iteración del filtro de Kalman extendido dual
```

Algoritmo 20 Ejemplo. Filtro de Kalman Extendido dual. Maquina síncrona. Continuación

```

 $xV(:, k) = x$  //almacenamiento de la estimación de estados
 $zeV(:, k) = Hx$  //almacenamiento de la salida o mediciones estimadas
 $\theta V(:, k) = \theta$  //almacenamiento de la estimación de parámetros
end for
cálculo de errores
 $MeasErr = zr - yV$  //error de medición
 $EstErr = zr - zeV$  //error de estimación
 $MeasErrCov = \text{sum}(MeasErrMeasErr)/\text{length}(MeasErr)$  //covarianza del error de medición
 $EstErrCov = \text{sum}(EstErrEstErr)/\text{length}(EstErr)$  //covarianza del error de estimación

```

Se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N que este indicado, en el proceso iterativo se calcula el estado real del sistema tru , el cual se describe de acuerdo con la ecuación 3.12, tru es almacenado en la matriz $truV$, a continuación se calcula el observador del sistema y , el cual se describe mediante la ecuación 3.13 y se almacena en la matriz yV , se calcula la salida real del sistema zr , la cual es igual a la salida medida sin tener en cuenta el ruido de la medición, esta salida se almacena en zrV .

Como paso siguiente se realiza una iteración del filtro de Kalman extendido dual, por ende como se explicó anteriormente se obtiene la estimación de estados x , la covarianza de estimación de estado P_x , la estimación de parámetros θ , la covarianza de estimación de parámetros P_θ , y la matriz de medición H , las estimaciones mencionadas se utilizan en la siguiente iteración como estimaciones iniciales, x es almacenada en la matriz xV , se obtiene la salida estimada del sistema ze , es decir el producto entre H y la estimación de estados x , esta salida es almacenada en zeV , la estimación de parámetros se almacena en la matriz θV , por último se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y las covarianzas de error $MeasErrCov$ y $EstErrCov$.

Una vez las iteraciones han terminado, las matrices de almacenamiento $truV$ y xV contienen todos los estados verdaderos y todos los estados estimados respectivamente, la estimación de estado se realiza únicamente para el primer estado del sistema, se puede observar en la figura 3.10 la matriz $truV$ (color verde) y la matriz xV (co-

lor azul), es posible deducir que el estado estimado se aproxima de manera considerable al estado verdadero respecto al primer estado del sistema, además se observa que la estimación de estados únicamente se realiza para el primer estado del sistema.

De igual manera, si las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas verdaderas y todas las salidas estimadas respectivamente, las matrices zmV y zeV son matrices de tamaño $(1 \times N)$ debido a que se tiene una sola salida en el sistema, por ende las salidas reales y estimadas serán las mismas para los cuatro estados del sistema, se puede observar en la figura 3.11 la matriz zmV (color rojo, puntos) y zeV (color magenta), es posible observar que las salidas estimadas se aproximan a las salidas medidas.

Finalmente en la figura 3.12 se puede observar la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y entre el error de estimación $EstErr$ (color azul), la figura muestra que el nivel de ruido es reducido una vez se ha implementado el filtro de Kalman, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, para el sistema de interés la covarianza de error antes de la implementación del filtro ($MeasErrCov$) es igual a 0,1397 y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 0,1091.

Es importante recordar que la estimación de parámetros corresponde a un proceso estacionario, por ende con los niveles de ruido utilizados en el ejemplo anterior, la estimación de parámetros es inestable, es necesario modificar las covarianzas Q_x y R_x , a un nivel de ruido muy pequeño, $Q_x = 0,1e - 35eye(n)$ y $R_x = 0,1e - 35$, con estos niveles de ruido la estimación de estados es perfecta, y la estimación de parámetros se estabiliza, de esta manera se encuentra el parámetro verdadero del sistema (figura 3.13).

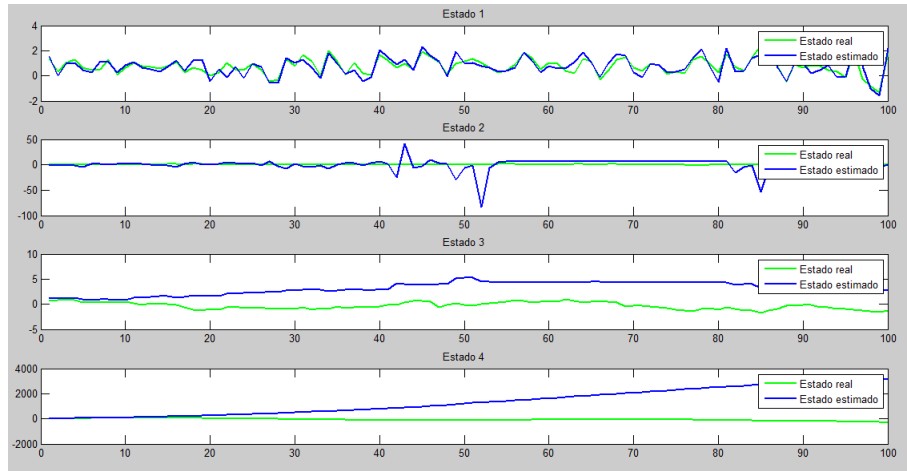


Figura 3.10: Estimación de estados mediante el filtro de Kalman extendido dual. Ejemplo (sistema de cuarto orden)

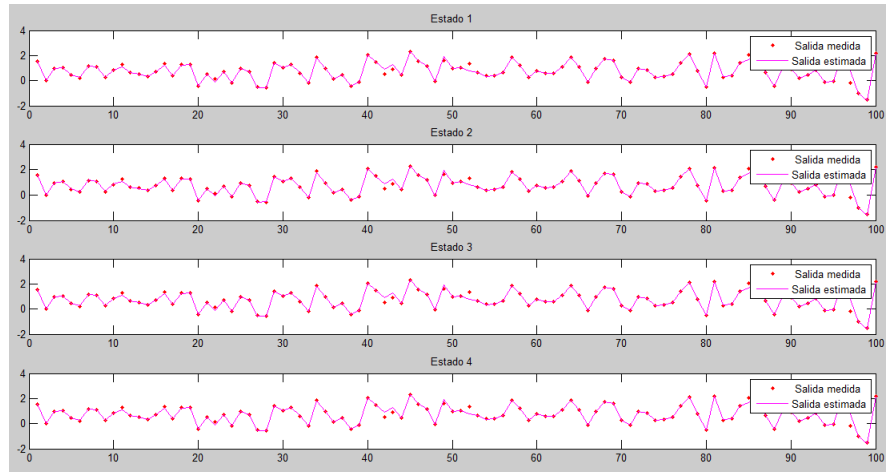


Figura 3.11: Salidas estimadas mediante el filtro de Kalman extendido dual. Ejemplo (sistema de cuarto orden)

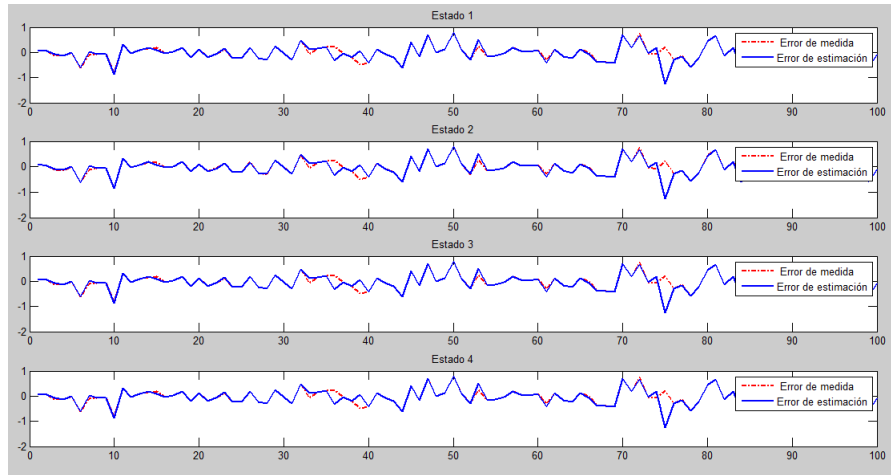


Figura 3.12: Error de estimación mediante el filtro de Kalman extendido dual. Ejemplo (sistema de cuarto orden)

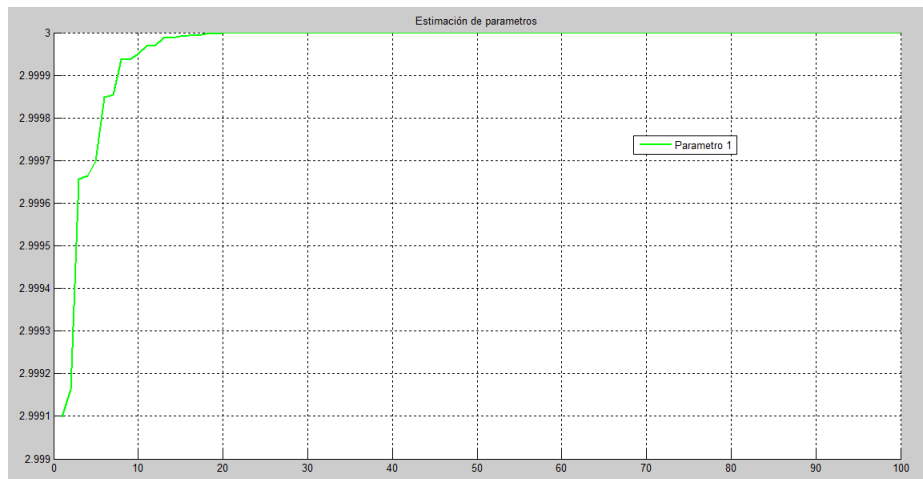


Figura 3.13: Estimación de parámetros mediante el filtro de Kalman extendido dual. Ejemplo (sistema de cuarto orden)

4. CONJUNTO DE FILTROS DE KALMAN

El filtro de Kalman da una solución óptima para la estimación de estados de un sistema lineal, cuando se trata de sistemas no lineales se emplea el filtro de Kalman extendido, el cual usa el operador jacobiano para linealizar el modelo del sistema, sin embargo el filtro de Kalman extendido presenta algunos inconvenientes, uno de ellos se encuentra en el manejo del error de estimación o error estadístico, el cual se considera como sustituto ya que corresponde al modelo del sistema linealizado y no al modelo del sistema original, además tanto en el filtro de Kalman como en el filtro de Kalman extendido es necesario almacenar y propagar en el tiempo la covarianza de estimación P haciendo a los métodos computacionalmente ineficientes para sistemas de alto orden [6][21].

El filtro de Kalman extendido también presenta un inconveniente relacionado con el uso del operador jacobiano, el cual puede no existir para algunos sistemas. En contraste con los métodos mencionados anteriormente el conjunto de filtros de Kalman representa el error de estimación o estadístico mediante un conjunto de modelos o realizaciones en lugar de emplear una expresión explícita, por consiguiente este método puede ser empleado con un costo computacional más bajo que el filtro de Kalman extendido [6][21].

Respecto a la elección adecuada del número de miembros del conjunto q , la literatura sobre este método sugiere que un conjunto de tamaño 50 a 100 es a menudo suficiente para sistemas con miles de estados. El conjunto de filtros de Kalman representa una alternativa en la estimación de estados de un sistema no lineal, hoy en día este método es ampliamente utilizado especialmente en predicción del clima, donde los modelos son no lineales y de orden muy alto [6][21].

Una descripción más detallada acerca del conjunto de filtros de Kalman puede encontrarse en [6] y [21], estas investigaciones se toman como referencia para presentar la deducción y las ecuaciones del algoritmo.

4.1. Estimación de estados

4.1.1. El problema de la estimación de estados

Se tiene el mismo problema de estimación de estados descrito anteriormente para el filtro de Kalman extendido, se busca estimar el modelo de espacio de estados de un sistema no lineal.

4.1.2. Deducción de ecuaciones

El método es presentado en tres etapas, en primer lugar se representa el error estadístico o error de estimación en los cálculos a-priori del sistema, de esta manera se tiene un conjunto de q estados estimados a-priori con errores de muestreo aleatorio, se denota el conjunto como \underline{x} de la forma.

$$\underline{x}_k = (\underline{x}_k^1, \dots, \underline{x}_k^q) \quad (4.1)$$

Se calcula la media de la estimación a-priori de la forma.

$$\overline{\underline{x}_k} = \frac{1}{q} \sum_{i=1}^q \underline{x}_k^i \quad (4.2)$$

Donde el termino \underline{x}_k^i representa el miembro i del conjunto estimado a-priori. Es necesario recordar algunas ecuaciones correspondientes al filtro de Kalman estándar, de esta manera en la ecuación 4.3 se observa el error de estimación representado por la covarianza de estimación de estados P .

$$\underline{P}_k = E[(x_k - \underline{x}_k)(x_k - \underline{x}_k)^T] \quad (4.3)$$

En el filtro de Kalman estándar la ganancia se calcula de la forma.

$$G_k = \underline{P}_k H_k^T [H_k \underline{P}_k H_k^T + R_k]^{-1} \quad (4.4)$$

Es posible expresar la ganancia de Kalman G mediante covarianzas.

$$G_k = (\underline{P}_{xy_k})(\underline{P}_{yy_k})^{-1} \quad (4.5)$$

Donde.

$$\underline{P}_{xy_k} = E[(x_k - \underline{x}_k)(y_k - \underline{y}_k)^T] = \underline{P}_k H_k^T \quad (4.6)$$

$$\underline{P}_{yy_k} = E[(y_k - \underline{y}_k)(y_k - \underline{y}_k)^T] = H_k \underline{P}_k H_k^T + R_k \quad (4.7)$$

Partiendo de que el estado del sistema x_k es desconocido se aproximan las ecuaciones 4.6 y 4.7 usando los miembros del conjunto. Se define el error del conjunto \underline{E}_k alrededor de la media del conjunto de la forma.

$$\underline{E}_k = (\underline{x}_k^1 - \underline{\bar{x}}_k, \dots, \underline{x}_k^q - \underline{\bar{x}}_k) \quad (4.8)$$

También se define el error de error de salida del conjunto \underline{E}_{y_k} de la forma.

$$\underline{E}_{y_k} = (\underline{y}_k^1 - \underline{\bar{y}}_k, \dots, \underline{y}_k^q - \underline{\bar{y}}_k) \quad (4.9)$$

Posteriormente se realizan las siguientes aproximaciones para las covarianzas del error de estimación \underline{P}_k , \underline{P}_{xy_k} y \underline{P}_{yy_k} .

$$\underline{P}_k = \frac{1}{q-1} \underline{E}_k (\underline{E}_k)^T \quad (4.10)$$

$$\underline{P}_{xy_k} = \frac{1}{q-1} \underline{E}_k (\underline{E}_{y_k})^T \quad (4.11)$$

$$\underline{P}_{yy_k} = \frac{1}{q-1} \underline{E}_{y_k} (\underline{E}_{y_k})^T \quad (4.12)$$

Se interpreta que la media del conjunto estimado a-priori es la mejor estimación a-priori de estados, además se interpreta que la propagación de los miembros del conjunto alrededor de la media es el error entre la mejor estimación y el estado actual del sistema. En segundo lugar se realiza el análisis respectivo para los cálculos a-posteriori del sistema, el conjunto de filtros de Kalman realiza un conjunto de ciclos de asimilación de datos en paralelo , donde $i = 1, \dots, q$.

$$x_k = \underline{x}_k^i + G_k(y_k^i - h(\underline{x}_k^i)) \quad (4.13)$$

El vector de observación está dado por.

$$y_k^i = y_k + v_k^i \quad (4.14)$$

Donde v_k^i representa el ruido de la medición, el cual se asume aditivo, blanco y gaussiano, con media cero y con matriz de covarianza R_k . El error de estimación a-posteriori está representado por \underline{P}_k .

$$P_k = \frac{1}{q-1} E_k (E_k)^T \quad (4.15)$$

Donde E_k está definido por la ecuación 4.8, con \underline{x}_k^i remplazado por x_k^i y \overline{x}_k remplazado por la media de los miembros del conjunto estimado a-posteriori. Se utiliza la expresión correspondiente a la ganancia del filtro de Kalman estándar empleando las covarianzas del error de estimación calculadas en las ecuaciones 4.11 y 4.12.

$$G_k = \underline{P}_k H_k^T [H_k \underline{P}_k H_k^T + R_k]^{-1} \quad (4.16)$$

Finalmente se realiza la predicción del error estadístico o error de estimación en los cálculos a-priori.

$$\underline{x}_{k+1} = f(x_k, u_k) + w_k^i \quad (4.17)$$

Donde w_k^i representa el ruido del proceso, el cual se asume aditivo, blanco y gaussiano, con media cero y con matriz de covarianza Q_k .

4.1.3. Ecuaciones para estimación de estados

- Cálculos a-priori

- Estimación del conjunto de estados a-priori

$$\underline{x}_{k+1} = f(x_k, u_k) + w_k^i \quad (4.18)$$

- Media de la estimación del conjunto de estados a-priori

$$\overline{x}_k = \frac{1}{q} \sum_{i=1}^q \underline{x}_k^i \quad (4.19)$$

- Error de estados en el conjunto

$$\underline{E}_k = (\underline{x}_k^1 - \overline{x}_k, \dots, \underline{x}_k^q - \overline{x}_k) \quad (4.20)$$

- Error de salida del conjunto

$$\underline{E}_{y_k} = (\underline{y}_k^1 - \overline{y}_k, \dots, \underline{y}_k^q - \overline{y}_k) \quad (4.21)$$

- Covarianzas de estimación del conjunto

$$\underline{P}_{xy_k} = \frac{1}{q-1} \underline{E}_k(\underline{E}_{y_k})^T \quad (4.22)$$

$$\underline{P}_{yy_k} = \frac{1}{q-1} \underline{E}_{y_k}(\underline{E}_{y_k})^T \quad (4.23)$$

- Ganancia de Kalman

$$G_k = (\underline{P}_{xy_k})(\underline{P}_{yy_k})^{-1} \quad (4.24)$$

- Calculos a-posteriori

- Estimación del conjunto de estados a-posteriori

$$x_k = \underline{x}_k^i + G_k(y_k^i - h(\underline{x}_k^i)) \quad (4.25)$$

- Media de la estimación del conjunto de estados a-posteriori

$$\overline{x_k} = \frac{1}{q} \sum_{i=1}^q x_k^i \quad (4.26)$$

4.1.4. Algoritmo para estimación de estados

Las ecuaciones del conjunto de filtros de kalman para estimación de estados se implementan en el algoritmo 21,22. (conjunto de filtros de Kalman para estimación de estados) tiene como objetivo estimar los estados de un sistema no lineal a partir de un conjunto de estados, se puede observar que el algoritmo necesita como entradas la matriz de transición de estados no lineal f , la matriz de medición no lineal h , el conjunto de estados estimado inicial x , el orden del sistema n , el número de miembros de conjunto q , el estado real del sistema tru y las covarianzas del proceso y medición respectivamente Q, R .

Anteriormente, para los cálculos a-priori del filtro de Kalman y el filtro de Kalman extendido era necesario definir un valor inicial de la estimación de estados x y de la covarianza de estimación P , para el conjunto de filtro de Kalman la estimación inicial de estados x está representada por un conjunto inicial de estados estimados x_{conj} .

Algoritmo 21 Conjunto de filtros de Kalman para estimación de estados

Require: f, h, x, Q, R, q, n, tru

Ensure: xm, zm_m, x

function $[xm, zm_m, x] = \text{conjunto de filtros de Kalman}(f, h, x, Q, R, q, n, tru)$
conjunto de estados

$x_{conj} = x$

matrices de almacenamiento

$\underline{x}_V = \text{zeros}(n, q)$ //almacenamiento de estimación del conjunto de estados a-priori

$\underline{y}_V = \text{zeros}(1, q)$ //almacenamiento del observador del conjunto de estados

$zm_V = \text{zeros}(1, q)$ //almacenamiento de la salida medida del sistema en el conjunto

$Ex_V = \text{zeros}(n, q)$ //almacenamiento del error de estados del conjunto

$Ey_V = \text{zeros}(1, q)$ //almacenamiento del error de salida del conjunto

cálculos a-priori del conjunto

for $j = 1 : q$ **do**

$w(:, j) = \text{sqr}(Q)\text{randn}(n, 1)$ //ruido de la evolución del conjunto de estados

$v(:, j) = \text{sqr}(R)\text{randn}(1, 1)$ //ruido de la medición del conjunto de estados

$\underline{x} = f(x_{conj}(:, j)) + w(:, j)$ //estimación del conjunto de estados a-priori

$zm = h(tru) + v(:, j)$ //salida medida del sistema real en el conjunto

$\underline{y} = h(\underline{x})$ //vector de observación del conjunto de estados

$\underline{x}_V(:, j) = \underline{x}$ //almacenamiento de estimación del conjunto de estados a-priori

$zm_V(:, j) = zm$ //almacenamiento de la salida medida del sistema en el conjunto

$\underline{y}_V(:, j) = \underline{y}$ //almacenamiento del observador del conjunto de estados

end for

cálculo de medias

$\underline{x}_m = \text{mean}(\underline{x}_V, 2)$ //media de la estimacion del conjunto de estados a-priori

$zm_m = \text{mean}(zm_V)$ //media de la salida medida del sistema en el conjunto

$\underline{y}_m = \text{mean}(\underline{y}_V)$ //media del vector de observacion del conjunto de estados

errores del conjunto

for $i = 1 : n$ **do**

$Ex = \underline{x}_V(i, :) - \underline{x}_m(i)$ //error de estados del conjunto

$Ex_V(i, :) = Ex$ //almacenamiento del error de estados del conjunto

end for

for $i = 1 : 1$ **do**

$Ey = \underline{y}_V(i, :) - \underline{y}_m(i)$ //error de salida del conjunto

$Ey_V(i, :) = Ey$ //almacenamiento del error de salida del conjunto

end for

Algoritmo 22 Conjunto de filtros de Kalman para estimación de estados. Continuación

covarianzas de estimación del conjunto

$$P_{xy} = Ex_V Ey_V' / (q - 1)$$

$$P_{yy} = Ey_V Ey_V' / (q - 1)$$

ganancia de Kalman

$$K = P_{xy} inv(P_{yy})$$

cálculos a-posteriori

$$x = \underline{x}_V + K * (zm_V - \underline{y}_V) // \text{estimación del conjunto de estados a-posteriori}$$

$$xm = mean(x, 2) // \text{media de la estimación del conjunto de estados a-posteriori}$$

return

Como se explicó anteriormente, este método no requiere almacenar la covarianza de estimación P , debido a que el error estadístico es expresado implícitamente por el conjunto de estados. Se definen cinco matrices de almacenamiento: \underline{x}_V almacenamiento de la estimación del conjunto de estados a-priori (\underline{x}), \underline{y}_V almacenamiento del vector de observación del conjunto de estados (\underline{y}), zm_V almacenamiento de la salida medida del sistema en el conjunto (zm), Ex_V almacenamiento del error de estados del conjunto (Ex), y finalmente Ey_V almacenamiento del error de salida del conjunto (Ey).

A continuación se da inicio a los cálculos a priori del conjunto, empieza un proceso iterativo desde $j = 1$ hasta el número total de miembros del conjunto q , en primer lugar se define un ruido de evolución de estados w y un ruido de medición de estados v para cada uno de los miembros del conjunto, luego se realiza la estimación a-priori de cada uno de los estados del conjunto \underline{x} , después se calcula la salida medida del sistema en el conjunto zm empleando los ruidos de medición del conjunto de estados v calculados anteriormente, de esta manera se garantiza que la matriz zm_v en la cual se almacena zm sea de tamaño $(1 \times q)$, se prosigue calculando el vector de observación del conjunto de estados \underline{y} evaluando los estados a-priori del conjunto \underline{x} en la función h , finalmente se almacenan los estados estimados a-priori del conjunto \underline{x} , la salida medida del sistema zm y el vector de observación del conjunto de estados \underline{y} en sus respectivas matrices.

Como paso siguiente se calculan las siguientes medias: media de la estimación del conjunto de estados a-priori \underline{x}_m , media de la salida medida del sistema en el conjunto zm_m , y media del vector de observación del conjunto de estados \underline{y} , luego se realiza un proceso iterativo desde $i = 1$ hasta el orden del sistema n , en el cual se calcula el error de es-

tados del conjunto Ex para ser almacenado en su respectiva matriz, además se realiza un proceso iterativo desde $i = 1$ hasta el número total de salidas del sistema 1 para calcular el error de salida del conjunto Ey para ser almacenado en su respectiva matriz.

Posteriormente se calculan las covarianzas de estimación del conjunto P_{xy} y P_{yy} empleando los errores de estados y de salida propios del conjunto calculados anteriormente, luego se calcula la ganancia de Kalman K , con la cual es posible realizar los cálculos a-posteriori del conjunto, se realiza la estimación del conjunto de estados a-posteriori y finalmente se calcula su respectiva media, debido a que la media de la estimación del conjunto de estados a-posteriori es la mejor estimación a-posteriori de estados.

Las salidas del algoritmo 21,22 son: en primer lugar la media de la estimación del conjunto de estados a-posteriori xm , la cual corresponde a la estimación del sistema, en segundo lugar la media de la salida medida del sistema zm_m , la cual corresponde a la salida medida del sistema real y finalmente la estimación del conjunto de estados a-posteriori x , la cual si se calcula en una iteración será empleada en la próxima como estimación inicial del conjunto de estados, en otras palabras x será almacenada y propagada en el tiempo, lo cual no ocurre con la covarianza de estimación P , la cual no se expresa de manera explícita, si no que se encuentra expresada implícitamente mediante el conjunto de estados.

4.1.5. Ejemplo 1. Estimación de estados

A continuación se presenta un ejemplo (algoritmo 23) en el cual se implementara el algoritmo 21,22 correspondiente al conjunto de filtros de Kalman para estimación de estados. El ejemplo describe un carro-péndulo modelado como un sistema de cuarto orden, la función f es tomada de la referencia [28].

$$f(x) = \begin{bmatrix} x(2) \\ x(2)^2 \sin x(1) \cos x(1) - \sin x(1) \\ x(4) \\ x(2)^2 \sin x(1) \end{bmatrix}$$

Para el modelo del sistema el estado uno corresponde al ángulo entre el péndulo y el eje vertical, el estado dos corresponde a la velocidad angular del péndulo, el estado tres corresponde al desplazamiento del auto y el estado cuatro corresponde a la velocidad del auto.

Algoritmo 23 Ejemplo 1. Conjunto de filtros de Kalman para estimación de estados.
Sistema SISO de cuarto orden

```

n = 4// orden del sistema
f=@(x)[x(2); x(2)x(2)sinx(1)cosx(1)-sinx(1); x(4); x(2)x(2)sinx(1)+sinx(1)cosx(1)]
//matriz de transición de estados no lineal
h = @(x)x(1)//matriz de medición no lineal
Q = diag([0,5^20,2^20,3^20,5^2])//ruido del proceso
R = 0,1//ruido de la medición
q = 50//numero de miembros del conjunto
x = ones(n,q)//conjunto inicial de estados estimados
mediciones o iteraciones
N = 80
    matrices de almacenamiento
    zrV = zeros(1, N)//almacenamiento de la salida real
    zmV = zeros(1, N)//almacenamiento de mediciones o salida medida
    zeV = zeros(1, N)//almacenamiento de la salida estimada
    xV = zeros(n, N)//almacenamiento de la estimación de estados
    truV = zeros(n, N)//almacenamiento del estado real del sistema
    estado real inicial
    tru = 1rand(n, 1)
    proceso iterativo
    for k = 1 : N do
        tru = f(tru) + sqrt(Q)randn(n, 1)//estado real del sistema
        truV(:, k) = tru //almacenamiento del estado real del sistema
        [xm, zm_m, x]= conjunto de filtros de Kalman (f, h, x, Q, R, q, n, tru)//realizar una
        iteración del conjunto de filtros de Kalman para estimación de estados
        zmV(1, k) = zm_m//almacenamiento de la salida medida del sistema real
        xV(:, k) = xm //almacenamiento de la estimación de estados
        ze = h(xm)//salida estimada del sistema
        zeV(:, k) = h(xm)//almacenamiento de la salida o mediciones estimadas
        zr = h(tru)//salida real del sistema
        zrV(1, k) = zr//almacenamiento de las salidas reales del sistema
    end for
    calculo de errores
    MeasErr = zr - zmV//error de medición
    EstErr = zr - zeV//error de estimación
    MeasErrCov = sum(MeasErrMeasErr)/length(MeasErr)//covarianza del error
    de medición
    EstErrCov = sum(EstErrEstErr)/length(EstErr)//covarianza del error de esti-
    mación

```

Se puede observar en el algoritmo 23 que en primer lugar se introducen las entradas que requiere el conjunto de filtros de Kalman f, h, x, Q, R, q, n, tru . (A diferencia del filtro de Kalman y el filtro de Kalman extendido, en los cuales el estado estimado inicial era un vector columna de tamaño $n \times 1$, en el conjunto de filtros de Kalman x está representado por un conjunto inicial de estados estimados de tamaño $n \times q$), como paso siguiente se define el número de iteraciones o mediciones N que se realizarán en el algoritmo, luego se definen cinco matrices de almacenamiento: zrV almacenamiento de la salida real del sistema, zmV almacenamiento de mediciones o salida medida, zeV almacenamiento de la salida estimada, xV almacenamiento de la estimación de estados, y finalmente $truV$ almacenamiento del estado real del sistema.

La función h permite elegir qué estado será estimado, de esta manera se elige h como la primera posición del vector de estados $x(1)$, lo anterior representa que la estimación será realizada de manera óptima únicamente para el primer estado del sistema, Para los estados restantes la estimación a-posteriori será igual a la a-priori $x = \underline{x}$, ya que solo se tiene ganancia de Kalman G en el primer estado.

Posteriormente se define un valor inicial del estado real del sistema tru , luego se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N que este indicado, se calcula el estado real del sistema tru de acuerdo con la ecuación 3.1, se prosigue con el almacenamiento de tru en la matriz $truV$, como paso siguiente se realiza una iteración del conjunto de filtros de Kalman, el cual entrega la estimación de estados xm , y la salida medida del sistema zm_m las cuales son almacenadas en sus respectivas matrices, además se tiene como salida la estimación del conjunto de estados a-posteriori x , la cual será empleada en la próxima iteración como estimación inicial del conjunto de estados.

Es necesario mencionar que a diferencia de los ejemplos correspondientes al filtro de Kalman y el filtro de Kalman extendido, en el conjunto de filtros de Kalman la salida medida del sistema se calcula dentro del propio algoritmo del conjunto de filtros de Kalman y no en el proceso iterativo del ejemplo, esto se debe a que el método necesita que zm sea de tamaño $1 \times q$ para realizar los cálculos a-posteriori, por ende se calcula la media de la salida medida del sistema en el conjunto zm_m , la cual corresponde a la verdadera salida medida del sistema.

A continuación se utiliza la estimación de estados xm para calcular la salida estimada del sistema ze , la cual se almacena en su respectiva matriz, además se calcula y se almacena la salida real del sistema, como último paso se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y las covarianzas de error $MeasErrCov$ y $EstErrCov$. Una vez las iteraciones han terminado, las matrices de almacenamiento $truV$ y xV contienen todos los estados verdaderos y todos los estados estimados respectivamente, como se explicó anteriormente la estimación de estado se realiza únicamente para el primer estado del sistema, de esta manera se puede observar en la figura 4.1 la matriz $truV$ (color verde) y la matriz xV (color azul), es posible deducir que los estados estimados coincide por completo con los estados verdadero para el primer estado del sistema.

De igual manera, si las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas verdaderas y todas las salidas estimadas respectivamente, las matrices zmV y zeV son matrices de tamaño $(1 \times N)$ debido a que se tiene una sola salida en el sistema, por ende las salidas reales y estimadas serán las mismas para los cuatro estados del sistema, se puede observar en la figura 4.2 la matriz zmV (color rojo, puntos) y zeV (color magenta), es posible deducir que las salidas estimadas coinciden por completo con las salidas medidas.

Finalmente en la figura 4.3 se puede observar la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y el error de estimación $EstErr$ (color azul), la figura muestra que el error de estimación es igual al error de medición, debido a que las salidas medidas del sistema coinciden por completo con las salidas estimadas, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, la covarianza de error antes de la implementación del filtro ($MeasErrCov$) y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 0,0021.

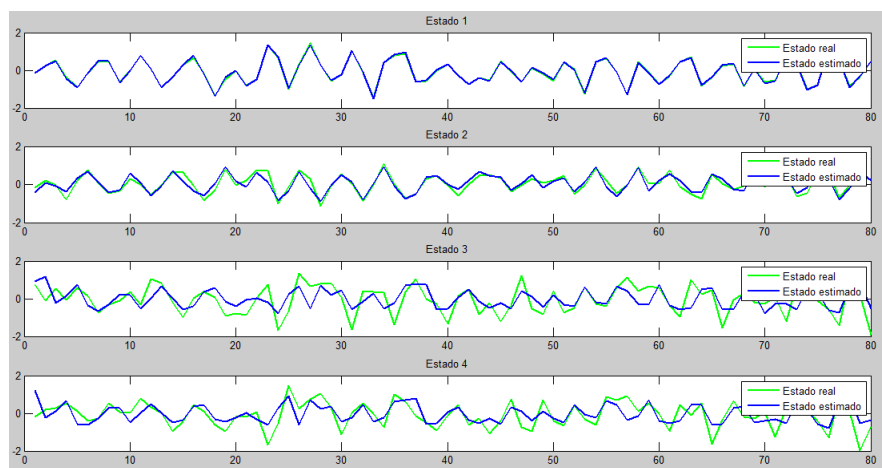


Figura 4.1: Estimación de estados mediante el conjunto de filtros de Kalman. Ejemplo 1 (sistema de cuarto orden)

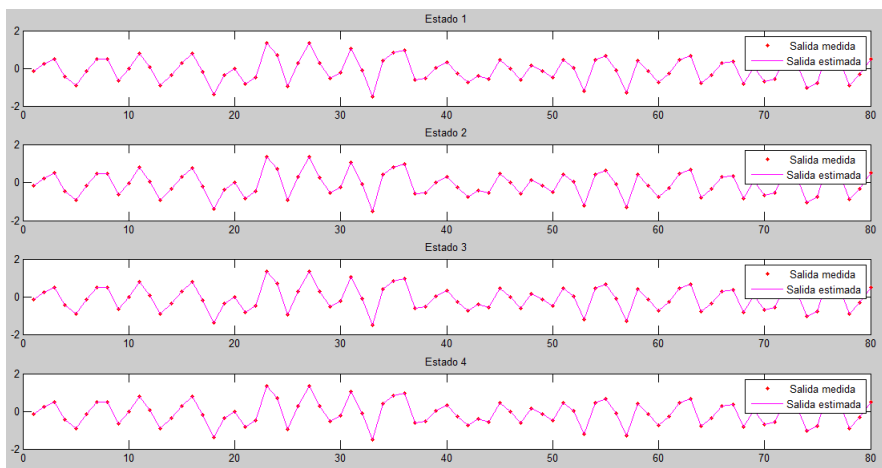


Figura 4.2: Salidas estimadas mediante el conjunto de filtros de Kalman. Ejemplo 1 (sistema de cuarto orden)

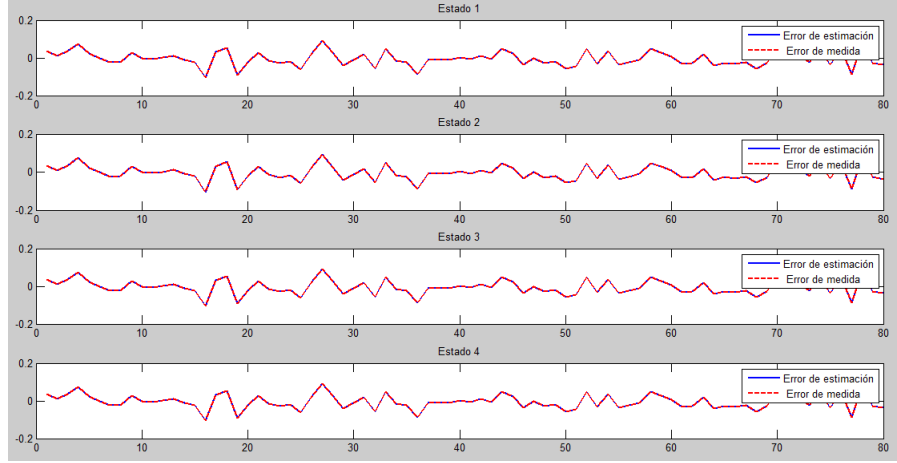


Figura 4.3: Error de estimación mediante el conjunto de filtros de Kalman. Ejemplo 1 (sistema de cuarto orden)

4.1.6. Ejemplo 2. Estimación de estados

A continuación se presenta un ejemplo (algoritmo 24) en el cual se implementa el algoritmo 21,22 correspondiente al conjunto de filtros de Kalman para estimación de estados. El ejemplo describe un sistema SISO (una entrada, una salida) de segundo orden y de comportamiento no lineal, la función f del ejemplo es tomada de la herramienta de MathWorks “learning ensemble Kalman filter”.

$$f(x) = \begin{bmatrix} x(1) + 0,1x(2) + 0,005 \\ x(2) + 0,1 \end{bmatrix}$$

Se puede observar en el algoritmo 24 que en primer lugar se introducen las entradas que requiere el conjunto de filtros de kalman, como paso siguiente se define el número de iteraciones o mediciones N que se realizaran en el algoritmo, luego se definen cinco matrices de almacenamiento. La función h permite elegir qué estado será estimado, de esta manera se elige h como la primera posición del vector de estados $x(1)$.

Posteriormente se define un valor inicial del estado real del sistema tru , luego se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N , luego se calcula el estado real del sistema tru , el cual se almacena en la matriz $truV$.

Algoritmo 24 Ejemplo 2. Conjunto de filtros de Kalman para estimación de estados. Sistema SISO de segundo orden

```

n = 2// orden del sistema
f = @(x)[x(1) + 0,1 * x(2) + 0,005; x(2) + 0,1]//matriz de transición de estados no
lineal
h = @(x)x(1)//matriz de medición no lineal
Q = Q = 0,1 * eye(n)//ruido del proceso
R = 1//ruido de la medición
q = 50//número de miembros del conjunto
x = ones(n, q)//conjunto inicial de estados estimados
mediciones o iteraciones
N = 80
    matrices de almacenamiento
    zrV = zeros(1, N)//almacenamiento de la salida real
    zmV = zeros(1, N)//almacenamiento de mediciones o salida medida
    zeV = zeros(1, N)//almacenamiento de la salida estimada
    xV = zeros(n, N)//almacenamiento de la estimación de estados
    truV = zeros(n, N)//almacenamiento del estado real del sistema
    estado real inicial
    tru = [1; 1]
    proceso iterativo
    for k = 1 : N do
        tru = f(tru) + sqrt(Q)randn(n, 1)//estado real del sistema
        truV(:, k) = tru //almacenamiento del estado real del sistema
        [xm, zm_m, x]= conjunto de filtros de Kalman (f, h, x, Q, R, q, n, tru)//realizar una
        iteración del conjunto de filtros de Kalman para estimación de estados
        zmV(1, k) = zm_m//almacenamiento de la salida medida del sistema real
        xV(:, k) = xm //almacenamiento de la estimación de estados
        ze = h(xm)//salida estimada del sistema
        zeV(:, k) = h(xm)//almacenamiento de la salida o mediciones estimadas
        zr = h(tru)//salida real del sistema
        zrV(1, k) = zr//almacenamiento de las salidas reales del sistema
    end for
    cálculo de errores
    MeasErr = zr - zmV//error de medición
    EstErr = zr - zeV//error de estimación
    MeasErrCov = sum(MeasErrMeasErr)/length(MeasErr)//covarianza del error
    de medición
    EstErrCov = sum(EstErrEstErr)/length(EstErr)//covarianza del error de esti-
    mación

```

Como paso siguiente se realiza una iteración del conjunto de filtros de Kalman, el cual entrega la estimación de estados xm , y la salida medida del sistema zm_m las cuales son almacenadas en sus respectivas matrices, además se tiene como salida la estimación del conjunto de estados a-posteriori x , la cual será empleada en la próxima iteración como estimación inicial del conjunto de estados.

A continuación se utiliza la estimación de estados xm para calcular la salida estimada del sistema ze , la cual se almacena en su respectiva matriz, además se calcula y se almacena la salida real del sistema, como último paso se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y las de error $MeasErrCov$ y $EstErrCov$. Una vez las iteraciones han terminado, las matrices de almacenamiento $truV$ y xV contienen todos los estados verdaderos y todos los estados estimados respectivamente, como se explicó anteriormente la estimación de estado se realiza únicamente para el primer estado del sistema, se puede observar en la figura 4.4 la matriz $truV$ (color verde) y la matriz xV (color azul), es posible deducir que los estados estimados coincide por completo con los estados verdadero para el primer estado del sistema.

De igual manera, si las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas verdaderas y todas las salidas estimadas respectivamente, las matrices zmV y zeV son matrices de tamaño $(1 \times N)$ debido a que se tiene una sola salida respecto a los dos estados del sistema, por ende las salidas reales y estimadas serán las mismas para los dos estados del sistema. De esta manera se puede observar en la figura 4.5 la matriz zmV (color rojo, puntos) y zeV (color magenta), es posible deducir que las salidas estimadas coinciden por completo con las salidas medidas.

Finalmente en la figura 4.6 se puede observar la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y entre el error de estimación $EstErr$ (color azul), la figura muestra que el error de estimación es igual al error de medición, debido a que las salidas medidas del sistema coinciden por completo con las salidas estimadas, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, la covarianza de error antes de la implementación del filtro ($MeasErrCov$) y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 0,0155.

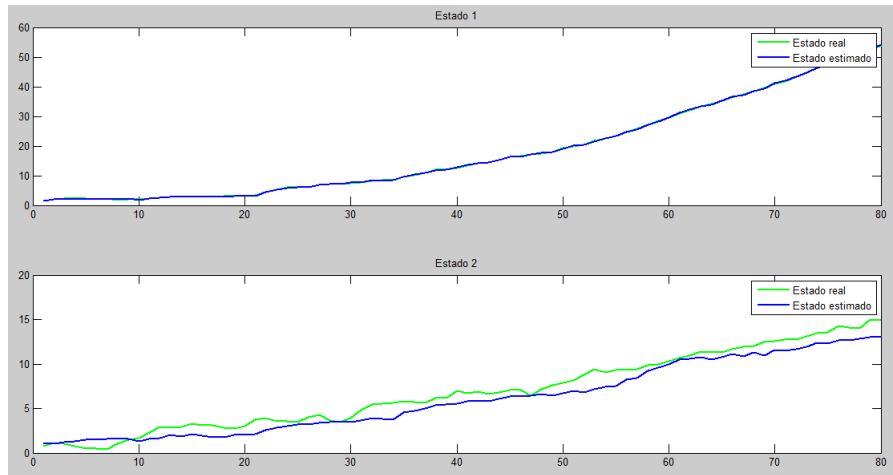


Figura 4.4: Estimación de estados mediante el conjunto de filtros de Kalman. Ejemplo 2 (sistema de segundo orden)

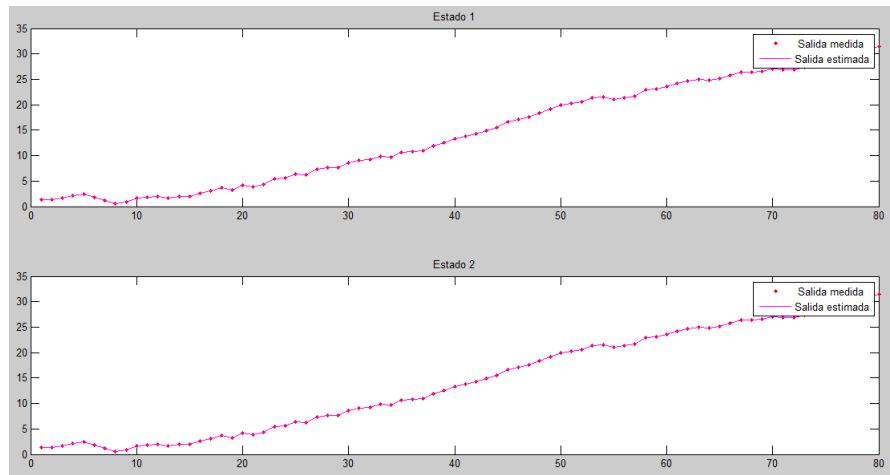


Figura 4.5: Salidas estimadas mediante el conjunto de filtros de Kalman. Ejemplo 2 (sistema de segundo orden)

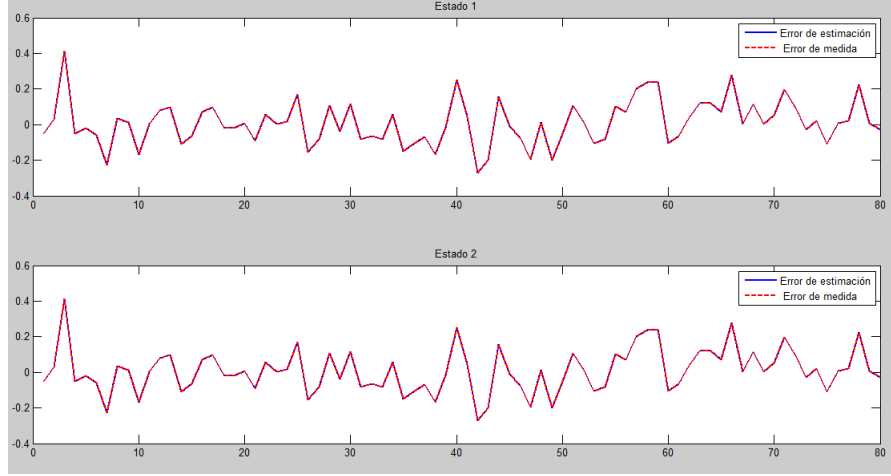


Figura 4.6: Error de estimación mediante el conjunto de filtros de Kalman. Ejemplo 2 (sistema de segundo orden)

4.2. Estimación de parámetros

El conjunto de filtros de Kalman también puede ser empleado para estimar los parámetros de un sistema no lineal. Una descripción más detallada acerca de la estimación de parámetros a partir del conjunto de filtros de Kalman puede encontrarse en [30] y [31], esta investigación se toma como referencia para definir las ecuaciones correspondientes a la estimación de parámetros mediante el filtro de Kalman extendido.

4.2.1. El problema de la estimación de parámetros

Se tiene el mismo problema de estimación de parámetros descrito anteriormente para el filtro de Kalman extendido, se busca estimar los parámetros de un sistema a partir de una función parametrizada no lineal.

4.2.2. Ecuaciones para estimación de parámetros

- Cálculos a-priori
 - Estimación del conjunto de parámetros a-priori

$$\underline{\theta}_{k+1} = \theta_k + w_k^i \quad (4.27)$$

- Media de la estimación del conjunto de parámetros a-priori

$$\overline{\theta_k} = \frac{1}{q} \sum_{i=1}^q \theta_k^i \quad (4.28)$$

- Error de parámetros en el conjunto

$$\underline{E_\theta} = (\underline{\theta_k}^1 - \overline{\theta_k} \dots \underline{\theta_k}^q - \overline{\theta_k}) \quad (4.29)$$

- Error de salida del conjunto

$$\underline{E_{y_k}} = (\underline{y_k}^1 - \overline{y_k} \dots \underline{y_k}^q - \overline{y_k}) \quad (4.30)$$

- Covarianzas de estimación del conjunto

$$\underline{P_{\theta y_k}} = \frac{1}{q-1} \underline{E_k} (\underline{E_{y_k}})^T \quad (4.31)$$

$$\underline{P_{yy_k}} = \frac{1}{q-1} \underline{E_{y_k}} (\underline{E_{y_k}})^T \quad (4.32)$$

- Ganancia de Kalman

$$G_k = (\underline{P_{\theta y_k}})(\underline{P_{yy_k}})^{-1} \quad (4.33)$$

- Calculos a-posteriori

- Estimación del conjunto de parámetros a-posteriori

$$\theta_k = \underline{\theta_k}^i + G_k(\underline{y_k}^i - h(\underline{\theta_k}^i)) \quad (4.34)$$

- Media de la estimación del conjunto de parámetros a-posteriori

$$\overline{\theta_k} = \frac{1}{q} \sum_{i=1}^q \theta_k^i \quad (4.35)$$

4.2.3. Algoritmo para estimación de parámetros

Las ecuaciones del conjunto de filtros de Kalman para estimación de parámetros se implementan en el algoritmo 25,26.

Algoritmo 25 Conjunto de filtros de Kalman para estimación de parámetros

Require: $G, \theta, Q, R, q, m, \theta_{tru}$ **Ensure:** θ_m, zm_m, θ **function** $[\theta_m, zm_m, \theta] = \text{conjunto de filtros de Kalman}(G, \theta, Q, R, q, m, \theta_{tru})$
conjunto de parámetros $\theta_{conj} = \theta$ *matrices de almacenamiento* $\underline{\theta}_V = \text{zeros}(m, q)$ //almacenamiento de estimación del conjunto de parámetros a-priori $\underline{y}_V = \text{zeros}(1, q)$ //almacenamiento del observador del conjunto de parámetros $zm_V = \text{zeros}(1, q)$ //almacenamiento de la salida medida del sistema en el conjunto $E\theta_V = \text{zeros}(n, q)$ //almacenamiento del error de parámetros del conjunto $Ey_V = \text{zeros}(1, q)$ //almacenamiento del error de salida del conjunto*cálculos a-priori del conjunto***for** $j = 1 : q$ **do** $w(:, j) = \text{sqr}(Q)\text{randn}(n, 1)$ //ruido de la evolución del conjunto de parámetros $v(:, j) = \text{sqr}(R)\text{randn}(1, 1)$ //ruido de la medición del conjunto de parámetros $\underline{\theta} = \theta_{conj}(:, j) + w(:, j)$ //estimación del conjunto de parámetros a-priori $zm = G(\theta_{tru}) + v(:, j)$ //salida medida del sistema real en el conjunto $\underline{y} = G(\underline{\theta})$ //vector de observación del conjunto de parámetros $\underline{\theta}_V(:, j) = \underline{\theta}$ //almacenamiento de estimación del conjunto de parámetros a-priori $zm_V(:, j) = zm$ //almacenamiento de la salida medida del sistema en el conjunto $\underline{y}_V(:, j) = \underline{y}$ //almacenamiento del observador del conjunto de parámetros**end for***cálculo de medias* $\underline{\theta}_m = \text{mean}(\underline{\theta}_V, 2)$ //media de la estimacion del conjunto de parámetros a-priori $zm_m = \text{mean}(zm_V)$ //media de la salida medida del sistema en el conjunto $\underline{y}_m = \text{mean}(\underline{y}_V)$ //media del vector de observacion del conjunto de parámetros*errores del conjunto***for** $i = 1 : m$ **do** $E\theta = \theta_V(i, :) - \underline{\theta}_m(i)$ //error de parámetros del conjunto $E\theta_V(i, :) = E\theta$ //almacenamiento del error de parámetros del conjunto**end for****for** $i = 1 : 1$ **do** $Ey = \underline{y}_V(i, :) - \underline{y}_m(i)$ //error de salida del conjunto $Ey_V(i, :) = Ey$ //almacenamiento del error de salida del conjunto**end for**

Algoritmo 26 Conjunto de filtros de Kalman para estimación de parámetros. Continuación

covarianzas de estimación del conjunto

$$P_{\theta y} = E\theta_V Ey_V' / (q - 1)$$

$$P_{yy} = Ey_V Ey_V' / (q - 1)$$

ganancia de Kalman

$$K = P_{\theta y} inv(P_{yy})$$

cálculos a-posteriori

$$\theta = \underline{\theta}_V + K * (zm_V - \underline{y}_V) // \text{estimación del conjunto de parámetros a-posteriori}$$

$$\theta m = mean(\theta, 2) // \text{media de la estimación del conjunto de parámetros a-posteriori}$$

return

El algoritmo 25,26 (conjunto de filtros de Kalman para estimación de parámetros) tiene como objetivo estimar los parámetros de una función parametrizada no lineal a partir de un conjunto de parámetros, se puede observar que el algoritmo necesita como entradas la función parametrizada no lineal G , el conjunto de parámetros estimados inicial θ , el número de parámetros m , el número de miembros de conjunto q , los parámetros verdaderos θ_{tru} y las covarianzas del proceso y medición respectivamente Q, R .

Anteriormente, para los cálculos a-priori del filtro de Kalman y el filtro de Kalman extendido era necesario definir un valor inicial de la estimación de parámetros θ y de la covarianza de estimación P , para el conjunto de filtro de Kalman la estimación inicial de parámetros θ está representada por un conjunto inicial de parámetros θ_{conj} , este método no requiere almacenar la covarianza de estimación P , debido a que el error estadístico es expresado implícitamente por el conjunto de parámetros.

Se definen cinco matrices de almacenamiento, a continuación se da inicio a los cálculos a priori del conjunto, empieza un proceso iterativo desde $j = 1$ hasta el número total de miembros del conjunto q , en primer lugar se define un ruido de evolución de parámetros w y un ruido de medición de parámetros v para cada uno de los miembros del conjunto, luego se realiza la estimación a-priori de cada uno de los parámetros del conjunto $\underline{\theta}$, después se calcula la salida medida del sistema en el conjunto zm empleando los ruidos de medición, se prosigue calculando el vector de observación del conjunto de parámetros \underline{y} evaluando los parámetros a-priori del conjunto $\underline{\theta}$ en la función G , finalmente se almacenan los parámetros estimados a-priori del conjunto $\underline{\theta}$, la salida medida del sistema zm y el vector de observación del conjunto de parámetros \underline{y} en sus respectivas matrices.

Como paso siguiente se calculan las siguientes medias: media de la estimación del conjunto de parámetros a-priori θ_m , media de la salida medida del sistema en el conjunto zm_m , y media del vector de observación del conjunto de parámetros y , luego se realiza un proceso iterativo desde $i = 1$ hasta el número de parámetros m , en el cual se calcula el error de parámetros del conjunto $E\theta$ para ser almacenado en su respectiva matriz, además se realiza un proceso iterativo desde $i = 1$ hasta el número total de salidas del sistema 1 para calcular el error de salida del conjunto Ey para ser almacenado en su respectiva matriz.

Posteriormente se calculan las covarianzas de estimación del conjunto $P_{\theta y}$ y P_{yy} empleando los errores de parámetros y de salida propios del conjunto calculados anteriormente, luego se calcula la ganancia de Kalman K , con la cual es posible realizar los cálculos a-posteriori del conjunto, se realiza la estimación del conjunto de parámetros a-posteriori y finalmente se calcula su respectiva media, debido a que la media de la estimación del conjunto de parámetros a-posteriori es la mejor estimación a-posteriori de parámetros.

Las salidas del algoritmo 25,26 son: en primer lugar la media de la estimación del conjunto de parámetros a-posteriori θ_m , en segundo lugar la media de la salida medida del sistema zm_m , y finalmente la estimación del conjunto de parámetros a-posteriori θ , la cual si se calcula en una iteración será empleada en la próxima como estimación inicial del conjunto de parámetros, en otras palabras θ será almacenada y propagada en el tiempo, lo cual no ocurre con la covarianza de estimación P , la cual no se expresa de manera explícita, si no que se encuentra expresada implícitamente mediante el conjunto de parámetros.

4.2.4. Ejemplo. Estimación de parámetros

Se presenta un ejemplo (algoritmo 27) en el cual se implementa el algoritmo 25,26 correspondiente al conjunto de filtros de Kalman para estimación de parámetros. En el ejemplo se presenta la función Rosenbrock [19].

$$f(x_1 + x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (4.36)$$

Se desea encontrar los valores de x_1 y x_2 que minimicen la función, de esta manera los valores óptimos de los parámetros son $x_1 = 1$ y $x_2 = 1$ [19].

Algoritmo 27 Ejemplo. Conjunto de filtros de Kalman para estimación de parámetros. Función Rosenbrock

```

m = 2//parámetros del sistema
G = @(theta)[100*((theta(2) - (theta(1)^2))^2)+(1 - theta(1))^2]//funcion Rosenbrock
Q = 0,1e - 35eye(m) //ruido del proceso
R = 0,1e - 35 //ruido de la medición
theta = rand(m, q)//conjunto inicial de estimación de parámetros
theta_tru = one(m, q)//parámetros reales
mediciones o iteraciones
N = 100
matrices de almacenamiento
zmV = zeros(1, N)//almacenamiento de mediciones o salida medida
zeV = zeros(1, N)//almacenamiento de la salida estimada
thetaV = zeros(m, N)//almacenamiento de los parámetros estimados (theta)
proceso iterativo
for k = 1 : N do
    [theta_m, zm_m, theta] = conjunto de filtros de Kalman(G, theta, Q, R, q, m, theta_tru)//realizar una
    iteración del conjunto de filtros de Kalman para estimación de parámetros
    zmV(1, k) = zm_m//almacenamiento de la salida medida del sistema
    thetaV(:, k) = theta_m //almacenamiento de la estimación de parámetros
    ze = G(theta_m)//salida estimada del sistema
    zeV(1, k) = ze//almacenamiento de la salida estimada
end for

```

Se puede observar en el algoritmo 27 que en primer lugar se introducen las entradas que requiere el conjunto de filtros de Kalman para estimación de parámetros $(G, \theta, Q, R, q, m, \theta_{tru})$, la estimación de parámetros inicial se representa mediante un conjunto de parámetros, además por facilidad de cálculo los parámetros reales también se representan mediante un conjunto. Como paso siguiente se define el número de iteraciones o mediciones N que se realizan en el algoritmo, luego se definen tres matrices de almacenamiento: zmV almacenamiento de la salida medida, zeV almacenamiento de la salida estimada, y finalmente θV almacenamiento de la estimación de parámetros.

A continuación se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta el número de iteraciones o mediciones N que este indicado, posteriormente se realiza una iteración del conjunto de filtros de Kalman para estimación de parámetros, por ende como se explicó anteriormente se obtiene la salida medida del sistema zm_m , la cual se almacena en zmV , también se obtiene la media de la estimación del conjunto de parámetros θm , la cual representa la mejor estimación de parámetros, θm se almacena en la matriz θV , se prosigue calculando la salida estimada de parámetros ze evaluando θm en la función parametrizada no lineal.

Es importante mencionar que para el caso de estimación de parámetros, el proceso tiene naturaleza estacionaria, por lo tanto el ruido del proceso y de medición son muy pequeños, pero no son cero, puesto que se afectaría el cálculo de la ganancia de kalman, en donde es necesario invertir la covarianza de estimación.

Una vez las iteraciones han terminado, la matriz de almacenamiento θV contienen todos los parámetros estimados, en la figura 4.7 es posible observar que para la iteración treinta los parámetros estimados coinciden de manera exacta con los parámetros reales, como se mencionó anteriormente, en este caso se estiman dos parámetros, parámetro 1 (color verde) y parámetro 2 (color azul).

De igual manera si las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas medidas y todas las salidas estimadas respectivamente, es importante destacar que las matrices mencionadas son de tamaño $1 \times N$, debido a que el sistema tiene una sola salida, en la figura 4.8 se observa la matriz zmV (color rojo, puntos) y la matriz zeV (color magenta), es posible deducir que las salidas coinciden de manera exacta una vez se han alcanzado los parámetros verdaderos.

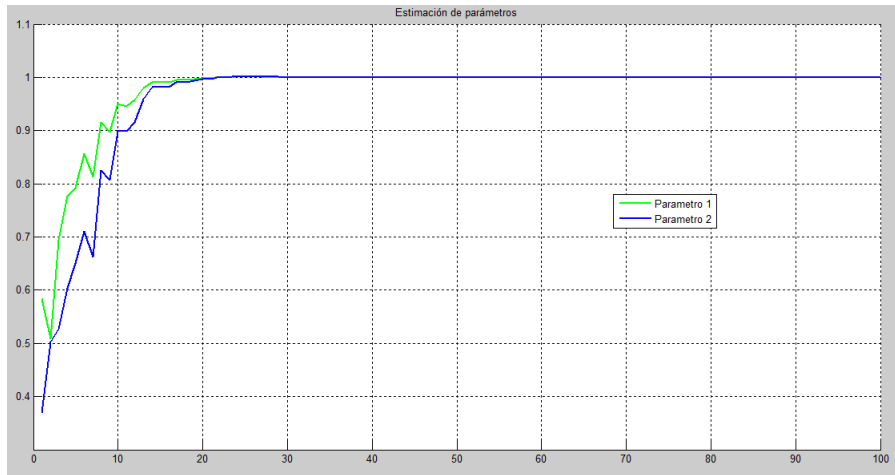


Figura 4.7: Estimación de parámetros mediante el conjunto de filtros de Kalman. Ejemplo (función Rosenbrock)

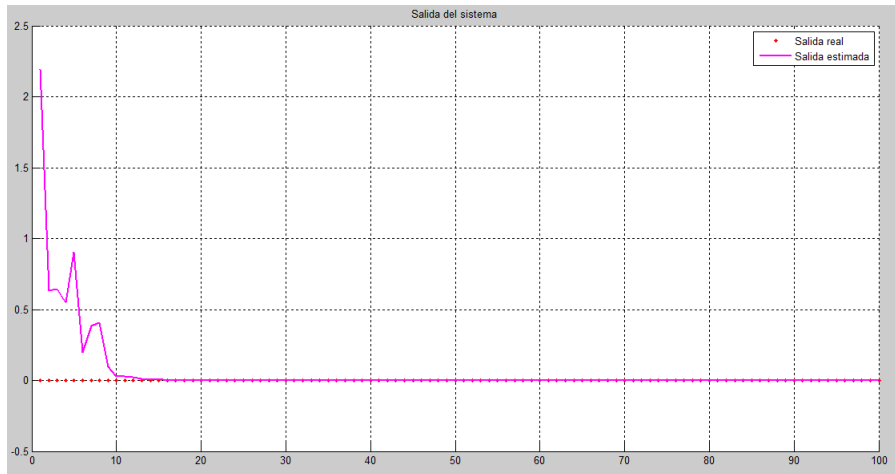


Figura 4.8: Salida estimada mediante el conjunto de filtros de Kalman. Ejemplo (función Rosenbrock)

4.3. Conjunto de filtros de Kalman dual

El conjunto de filtros de Kalman dual requiere dos representaciones en espacio de estados separadas para la estimación de estados y parámetros, por ende se emplean dos algoritmos de manera paralela [30].

4.3.1. El problema de la estimación dual

Para la estimación de estados se considera el siguiente modelo en espacio de estados:

$$x_{k+1} = f(x_k, u_k, \theta) + v_{x_k} \quad (4.37)$$

$$y_{x_k} = h(x_k, \theta) + n_{x_k} \quad (4.38)$$

w_{x_k} y v_{x_k} corresponden al ruido del proceso y al ruido de la medición para estimación de estados, los ruidos se asumen aditivos, blancos y gaussianos, con media cero y con matriz de covarianza Q_{x_k} y R_{x_k} respectivamente. Para la estimación de parámetros se considera el siguiente modelo en espacio de estados:

$$\theta_k = \theta_{k-1} + w_{\theta_k} \quad (4.39)$$

$$y_{\theta_k} = h(\theta_k) + v_{\theta_k} \quad (4.40)$$

w_{θ_k} y v_{θ_k} corresponden al ruido del proceso y al ruido de la medición para estimación de parámetros, los ruidos se asumen aditivos, blancos y gaussianos, con media cero y con matriz de covarianza Q_{θ_k} y R_{θ_k} respectivamente.

4.3.2. Ecuaciones para estimación dual

Para el conjunto de filtros de Kalman dual se emplean las ecuaciones para estimación de estados y para estimación de parámetros descritas en las subsecciones 4.1.3 y 4.2.2 respectivamente.

4.3.3. Algoritmo para estimación dual

Las ecuaciones del conjunto de filtros de Kalman dual para estimación de estados y parámetros se implementan en el algoritmo 28,29,30.

Algoritmo 28 Conjunto de filtros de Kalman dual

Require: $f, h, x, Q_x, R_x, q_x, n, tru, \theta, Q_\theta, R_\theta, q_\theta, m, \theta mtru$

Ensure: $xm, zm_m, x, \theta_m, \theta$

function $[xm, zm_m, x, \theta_m, \theta]$ = conjunto de filtros de Kalman dual
($f, h, x, Q_x, R_x, q_x, n, tru, \theta, Q_\theta, R_\theta, q_\theta, m, \theta mtru$)

matrices de almacenamiento

$\underline{x}_V = \text{zeros}(n, q)$ //almacenamiento de estimación del conjunto de estados a-priori

$\underline{y}_V = \text{zeros}(1, q)$ //almacenamiento del observador del conjunto de estados

$zm_V = \text{zeros}(1, q)$ //almacenamiento de la salida medida del sistema en el conjunto

$Ex_V = \text{zeros}(n, q)$ //almacenamiento del error de estados del conjunto

$Ey_V = \text{zeros}(1, q)$ //almacenamiento del error de salida del conjunto

$\underline{\theta}_V = \text{zeros}(m, q)$ //almacenamiento de la estimación del conjunto de parámetros a-priori

$\underline{y}_\theta V = \text{zeros}(1, q)$ //almacenamiento del observador del conjunto de parámetros

$zm_\theta V = \text{zeros}(1, q)$ //almacenamiento de la salida medida para parámetros

$E_\theta V = \text{zeros}(n, q)$ //almacenamiento del error de parámetros del conjunto

$Ey_\theta V = \text{zeros}(1, q)$ //almacenamiento del error de salida para parámetros

$x_{conj} = x$ //conjunto de estados

$\theta_{conj} = \theta$ //conjunto de parámetros

estimación de estados

cálculos a-priori para estimación de estados

for $j = 1 : q_x$ **do**

$w(:, j) = \text{sqr}(Q_x)\text{randn}(n, 1)$ //ruido de la evolución del conjunto de estados

$v(:, j) = \text{sqr}(R_x)\text{randn}(1, 1)$ //ruido de la medición del conjunto de estados

$\underline{x} = f(x_{conj}(:, j), \theta mtru) + w(:, j)$ //estimación del conjunto de estados a-priori

$zm = h(tru) + v(:, j)$ //salida medida del sistema real en el conjunto

$\underline{y} = h(\underline{x})$ //vector de observación del conjunto de estados

$\underline{x}_V(:, j) = \underline{x}$ //almacenamiento de estimación del conjunto de estados a-priori

$zm_V(:, j) = zm$ //almacenamiento de la salida medida del sistema en el conjunto

$\underline{y}_V(:, j) = \underline{y}$ //almacenamiento del observador del conjunto de estados

end for

cálculo de medias

$\underline{x}_m = \text{mean}(\underline{x}_V, 2)$ //media de la estimacion del conjunto de estados a-priori

$zm_m = \text{mean}(zm_V)$ //media de la salida medida del sistema en el conjunto

$\underline{y}_m = \text{mean}(\underline{y}_V)$ //media del vector de observacion del conjunto de estados

errores del conjunto

for $i = 1 : n$ **do**

$Ex = \underline{x}_V(i, :) - \underline{x}_m(i)$ //error de estados del conjunto

$E_{xV}(i, :) = Ex$ //almacenamiento del error de estados del conjunto

end for

Algoritmo 29 Conjunto de filtros de Kalman dual. Continuación

```
for  $i = 1 : 1$  do
     $Ey = y_V(i, :) - y_m(i)$  //error de salida del conjunto
     $Ey_V(i, :) = Ey$  //almacenamiento del error de salida del conjunto
end for
covarianzas de estimación del conjunto
 $P_{xy} = Ex_V Ey_V' / (q - 1)$ 
 $P_{yy} = Ey_V Ey_V' / (q - 1)$ 
ganancia de Kalman para estimación de estados
 $K = P_{xy} inv(P_{yy})$ 
cálculos a-posteriori para estimación de estados
 $x = x_V + K(zm_V - y_V)$  //estimación del conjunto de estados a-posteriori
 $xm = mean(x, 2)$  //media de la estimación del conjunto de estados a-posteriori
estimación de parámetros
cálculos a-priori para estimación de parámetros
for  $j = 1 : q_\theta$  do
     $w_\theta(:, j) = sqrt(Q_\theta) randn(m, 1)$  //ruido del proceso del conjunto de parámetros
     $v_\theta(:, j) = sqrt(R_\theta) randn(1, 1)$  //ruido de la medición del conjunto de parámetros
     $\underline{\theta} = \theta_{conj}(:, j) + w_\theta(:, j)$  //estimación del conjunto de parámetros a-priori
     $zm_\theta = h(\theta_{tru}) + v_\theta(:, j)$  //salida medida del sistema en el conjunto
     $y_\theta = h(\underline{\theta})$  //vector de observación del conjunto de parámetros
     $\underline{\theta}_V(:, j) = \underline{\theta}$  //almacenamiento de estimación del conjunto de parámetros a-priori
     $zm_\theta V(:, j) = zm_\theta$  //almacenamiento de la salida medida en el conjunto
     $y_\theta V(:, j) = y_\theta$  //almacenamiento del observador del conjunto de parámetros
end for
cálculo de medias
 $\underline{\theta}_m = mean(\underline{\theta}_V, 2)$  //media de la estimación del conjunto de parámetros a-priori
 $y_\theta m = mean(y_\theta V)$  //media del vector de observación del conjunto de parámetros
errores del conjunto
for  $i = 1 : m$  do
     $E\theta = \theta_V(i, :) - \underline{\theta}_m(i)$  //error de parámetros del conjunto
     $E\theta_V(i, :) = E\theta$  //almacenamiento del error de parámetros del conjunto
end for
for  $i = 1 : 1$  do
     $Ey_\theta = y_\theta V(i, :) - y_\theta m(i)$  //error de salida del conjunto
     $Ey_\theta V(i, :) = Ey_\theta$  //almacenamiento del error de salida del conjunto
end for
return
```

Algoritmo 30 Conjunto de filtros de Kalman dual. Continuación dos

covarianzas de estimación del conjunto

$$P_{\theta y} = E\theta_V E y_\theta V' / (q_\theta - 1)$$

$$P_{yy_\theta} = E y_\theta V E y_\theta V' / (q_\theta - 1)$$

ganancia de Kalman para estimación de parámetros

$$K = P_{\theta y} \text{inv}(P_{yy_\theta})$$

cálculos a-posteriori

$$\theta = \theta_V + K(zm_\theta V - y_\theta V) // \text{estimación del conjunto de parámetros a-posteriori}$$

$$\theta m = \text{mean}(\theta, 2) // \text{media de la estimación del conjunto de parámetros a-posteriori}$$

return

El algoritmo 28,29,30 (conjunto de filtros de Kalman dual) tiene como objetivo estimar los estados y los parámetros de un sistema no lineal, las entradas del algoritmo corresponden a las utilizadas para estimación de estados y para estimación de parámetros, se puede observar que la función f esta en terminos de los estados x y los parámetros θ , para la estimación de estados se emplean los parámetros verdaderos del sistema, por otro lado para estimación de parámetros la salida medida se obtiene evaluando los parámetros verdaderos del sistema en la función de salida h .

4.3.4. Ejemplo. Estimación dual

A continuación se presenta un ejemplo (algoritmo 31) en el cual se implementa el algoritmo 28,29,30 correspondiente al conjunto de filtros de Kalman dual. El ejemplo describe una maquina síncrona modelada como un sistema de cuarto orden, la función f es tomada de la referencia [29].

$$f(x) = \begin{bmatrix} x(1) - 10(\theta x(1) - 3,1416x(3)\sin(x(4)) - 2)/30 \\ x(2) - 10(\theta x(2) + 3,1416x(3)\cos(x(4)) - 2)/30 \\ x(3) \\ x(4) + 10x(3) \end{bmatrix}$$

Para el modelo del sistema el estado uno y el estado dos corresponden a corrientes del sistema, el estado tres corresponde a la frecuencia de operación y el estado cuatro corresponde al desfase de la máquina; el parámetro del sistema corresponde a una resistencia.

Algoritmo 31 Ejemplo. Conjunto de filtros de Kalman dual. Maquina síncrona

$n = 4$ // orden del sistema
 $m = 1$ // parámetros del sistema
 $f = @(x, \theta m)[x(1) - 10(\theta m x(1) - 3,1416x(3)\sin(x(4)) - 2)/30; x(2) - 10(\theta m x(2) + 3,1416x(3)\cos(x(4)) - 2)/30; x(3); x(4) + 10x(3)]$ //matriz de transición de estados
no lineal
 $h = @(x)x(1)$ //salida del sistema.
datos para estimación de estados
 $Q_x = 0,1eye(n)$ //ruido del proceso
 $R_x = 0,1$ //ruido de la medición
 $q_x = 20$ //miembros del conjunto
 $x = ones(n, q_x)$ //conjunto de estados
 $tru = [1; 1; 1; 1]$ //estado real del sistema
datos para estimación de parámetros
 $Q_\theta = 0,1e - 30eye(m)$ //ruido del proceso
 $R_\theta = 0,1e - 30$ //ruido de la medición
 $q_\theta = 20$ //miembros del conjunto
 $\theta = 0,1e - 30rand(m, q_\theta)$ //conjunto de parámetros
 $\theta mtru = 3$ //parámetro real
mediciones o iteraciones
 $N = 100$
matrices de almacenamiento
 $zrV = zeros(1, N)$ //almacenamiento de la salida real
 $zmV = zeros(1, N)$ //almacenamiento de mediciones o salida medida
 $zeV = zeros(1, N)$ //almacenamiento de la salida estimada
 $xV = zeros(n, N)$ //almacenamiento de la estimación de estados
 $truV = zeros(n, N)$ //almacenamiento del estado real del sistema
 $\theta V = zeros(m, N)$ //almacenamiento de la estimación de parámetros
proceso iterativo
for $k = 1 : N$
 $tru = f(tru, \theta mtru) + sqrt(Q_x)randn(n, 1)$ //estado real del sistema
 $truV(:, k) = tru$ //almacenamiento del estado real del sistema
 $[xm, zm_m, x, \theta_m, \theta] =$ conjunto de filtros de Kalman dual
 $(f, h, x, Q_x, R_x, q_x, n, tru, \theta, Q_\theta, R_\theta, q_\theta, m, \theta mtru)$ //realizar una iteración
 $zmV(1, k) = zm_m$ //almacenamiento de la salida medida del sistema real
 $xV(:, k) = xm$ //almacenamiento de la estimación de estados
 $ze = h(xm)$ //salida estimada del sistema
 $zeV(:, k) = h(xm)$ //almacenamiento de la salida o mediciones estimadas

Algoritmo 32 Ejemplo. Filtro de Kalman Extendido dual. Maquina síncrona. Continuación

```

 $zr = h(tru)$  //salida real del sistema
 $zrV(1, k) = zr$  //almacenamiento de las salidas reales del sistema
 $\theta V(:, k) = \theta m$  //almacenamiento de la estimación de parámetros
end for
cálculo de errores
 $MeasErr = zr - zmV$  //error de medición
 $EstErr = zr - zeV$  //error de estimación
 $MeasErrCov = sum(MeasErrMeasErr)/length(MeasErr)$  //covarianza del error de medición
 $EstErrCov = sum(EstErrEstErr)/length(EstErr)$  //covarianza del error de estimación

```

Se puede observar en el algoritmo 31,32 que en primer lugar se introducen las entradas que requiere el conjunto de filtros de Kalman dual $f, h, x, Q_x, R_x, q_x, n, tru, \theta, Q_\theta, R_\theta, q_\theta, m, \theta m, tru$, se define el estado real del sistema tru , y el número de iteraciones N que se realizan en el algoritmo, luego se definen seis matrices de almacenamiento.

Se da paso al proceso iterativo, el cual se realiza desde $k = 1$ hasta N , se calcula el estado real del sistema tru , el cual se describe de acuerdo con la ecuación 4.37, tru es almacenado en la matriz $truV$, como paso siguiente se realiza una iteración del conjunto de filtros de Kalman dual, por ende se obtiene la estimación de estados xm , y la salida medida del sistema zm_m las cuales son almacenadas en sus respectivas matrices, además se tiene como salida la estimación del conjunto de estados a-posteriori x , por otra parte se obtiene la estimación de parámetros θm , la cual se almacena en su respectiva matriz y además se obtiene la estimación del conjunto de parámetros a-posteriori θ , los conjuntos estimados son empleados en la siguiente iteración como conjuntos iniciales.

Posteriormente se obtiene la salida estimada del sistema ze , evaluando xm en la función h , luego se obtiene la salida real zr , evaluando tru en la función h , ze y zr son almacenadas, por último se calcula el error de medición $MeasErr$, el error de estimación $EstErr$ y las covarianzas de error $MeasErrCov$ y $EstErrCov$. Una vez las iteraciones han terminado, las matrices de almacenamiento $truV$ y xV contienen todos los estados verdaderos y todos los estados estimados respectivamente, la estimación de estado se

realiza para el primer estado del sistema, se puede observar en la figura 4.9 la matriz $truV$ (color verde) y la matriz xV (color azul), es posible deducir que el estado estimado se aproxima de manera optima al estado verdadero respecto al primer estado del sistema.

De igual manera, si las iteraciones han terminado, las matrices de almacenamiento zmV y zeV contienen todas las salidas verdaderas y todas las salidas estimadas respectivamente, las salidas reales y estimadas son las mismas para los cuatro estados del sistema, se puede observar en la figura 4.10 la matriz zmV (color rojo, puntos) y zeV (color magenta), es posible observar que las salidas estimadas coinciden casi por completo con las salidas medidas.

En la figura 4.11 se puede observar la comparación entre el error de medición $MeasErr$ (color rojo, línea discontinua) y el error de estimación $EstErr$ (color azul), la figura muestra que el error de estimación es igual al error de medición, debido a que las salidas medidas del sistema coinciden casi por completo con las salidas estimadas, lo anterior es confirmado mediante el cálculo de la covarianza de los errores, la covarianza de error antes de la implementación del filtro ($MeasErrCov$) y la covarianza de error después de implementado el filtro ($EstErrCov$) es igual a 0,0056.

Finalmente cabe recordar que la estimación de parámetros corresponde a un proceso estacionario, por ende se utiliza un nivel de ruido muy pequeño, $Q_\theta = 0,1e - 30eye(m)$ y $R_\theta = 0,1e - 30$, con estos niveles de ruido la estimación de parámetros se estabiliza, de esta manera se encuentra el parámetro verdadero del sistema(figura 4.12).

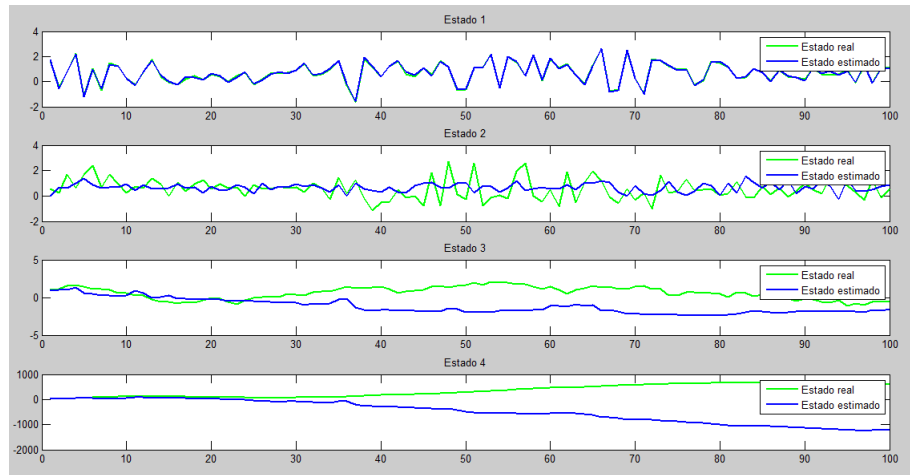


Figura 4.9: Estimación de estados mediante el conjunto de filtros de Kalman dual. Ejemplo (sistema de cuarto orden)

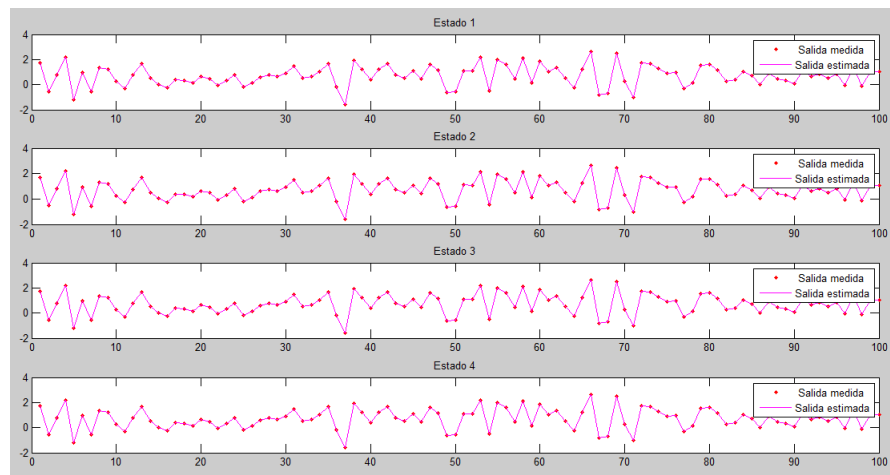


Figura 4.10: Salidas estimadas mediante el conjunto de filtros de Kalman dual. Ejemplo (sistema de cuarto orden)

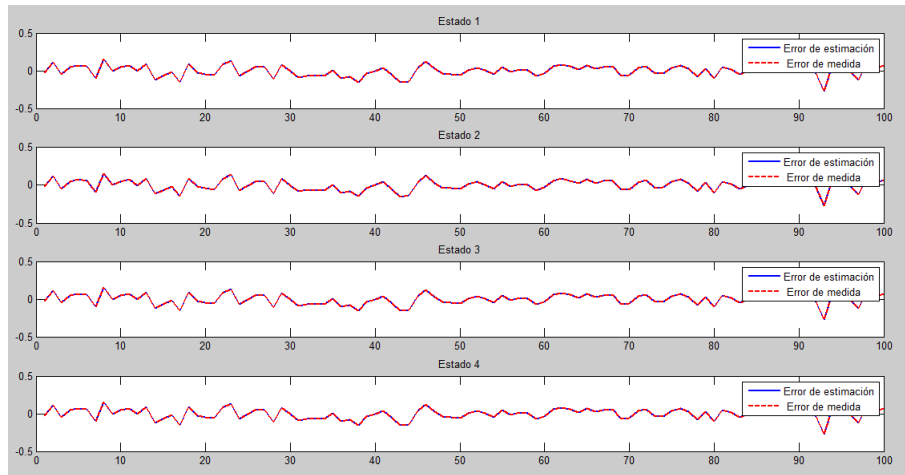


Figura 4.11: Error de estimación mediante el conjunto de filtros de Kalman dual. Ejemplo (sistema de cuarto orden)

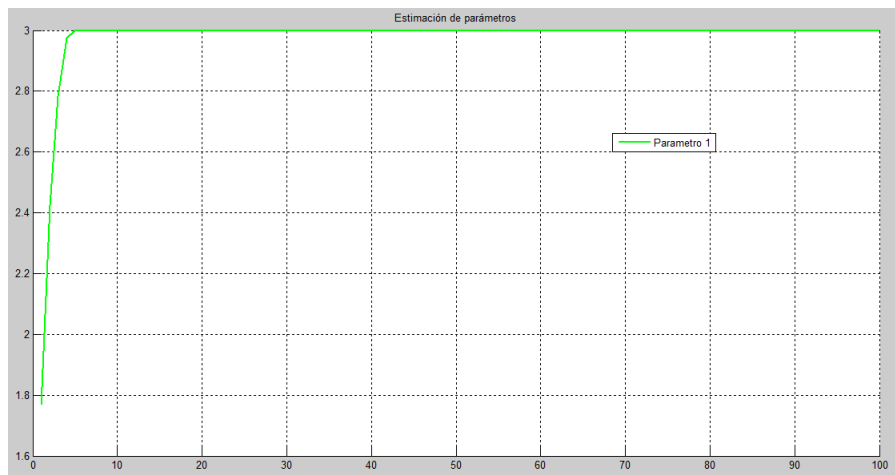


Figura 4.12: Estimación de parámetros mediante el conjunto de filtros de Kalman dual. Ejemplo (sistema de cuarto orden)

5. CONTROL DE UN SISTEMA DE GENERACIÓN EÓLICA A PARTIR DEL CONJUNTO DE FILTROS DE KALMAN

A continuación se busca diseñar un controlador para un sistema eólico a partir de los estados estimados por el conjunto de filtros de Kalman, el sistema a controlar corresponde a un generador de inducción-red eléctrica, el cual es un subsistema de una WECS (sistema de conversión de energía eólica). En la figura 5.1 es posible observar la turbina eólica acoplada con la red eléctrica, la representación y el modelo del sistema se toman de la referencia [32].

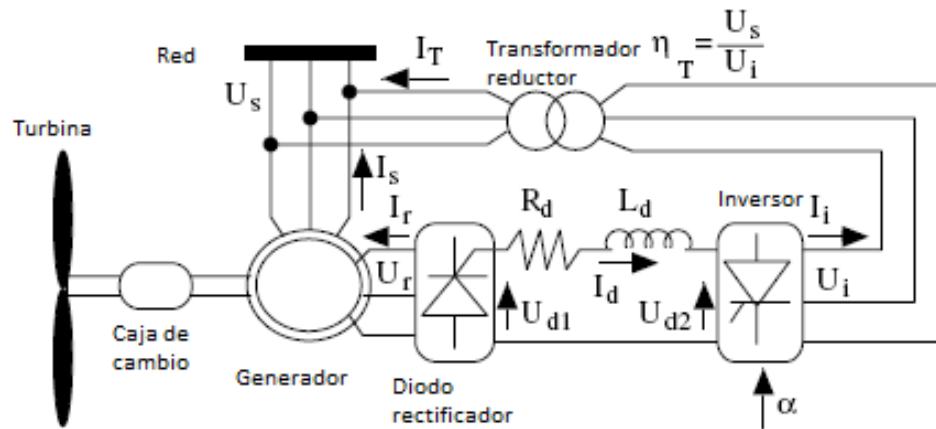


Figura 5.1: Sistema de generación eólica (generador de inducción-red eléctrica)

5.1. Modelo del sistema eólico

Se presenta el modelo del generador de inducción-red eléctrica, las ecuaciones del voltaje del rotor y el estator son simuladas por un modelo de cuarto orden, la representación en espacio de estados de la maquina con las corrientes como variables de estado se muestra a continuación:

$$\begin{bmatrix} \dot{i}_{sd} \\ \dot{i}_{sq} \\ \dot{i}_{rd} \\ \dot{i}_{rq} \end{bmatrix} = \frac{w_0}{B} \begin{bmatrix} -r_s X_r & (wB + w_t X_m^2) & -r_r X_m & -w_t X_r X_m \\ -(wB + w_t X_m^2) & -r_s X_r & w_t X_r X_m & -r_r X_m \\ -r_s X_m & w_t X_s X_m & -r_r X_s & (wB - w_t X_s X_r) \\ -w_t X_s X_m & -r_s X_m & -(wB - w_t X_s X_r) & -r_r X_s \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \\ i_{rq} \end{bmatrix} + \frac{w_0}{B} \begin{bmatrix} -X_r & 0 & X_m & 0 \\ 0 & -X_r & 0 & X_m \\ -X_m & 0 & X_s & 0 \\ 0 & -X_m & 0 & X_s \end{bmatrix} \begin{bmatrix} u_{sd} \\ u_{sq} \\ u_{rd} \\ u_{rq} \end{bmatrix}$$

Donde d y q representan devanados, u_{sd} y u_{sq} son voltajes del estator, i_{sd} , i_{sq} , i_{rd} e i_{rq} son las corrientes en el estator y el rotor, X_s y X_r son las reactancias del estator y el rotor, X_m es la reactancia de magnetización, w es una frecuencia angular arbitraria, w_t es la frecuencia angular del rotor, $w_o = 2\pi f$ es la frecuencia angular base, f_0 es la frecuencia de red, finalmente los voltajes en el rotor u_{rd} y u_{rq} son iguales a cero.

5.2. Algoritmo para estimación de estados

Las ecuaciones del conjunto de filtros de Kalman para estimación de estados implementadas en el algoritmo 21,22 deben ser modificadas, de tal manera que puedan ser empleadas en un modelo lineal.

El algoritmo 33,34 (conjunto de filtros de Kalman para estimación de estados de un sistema lineal), a diferencia del algoritmo 21,22 empleado para sistemas no lineales, necesita la matriz de transición de estados A , y la matriz de medición H en lugar de emplear las funciones f y h , las cuales corresponden a una representación en espacio de estados no lineal, además en este caso se necesita la matriz de entrada B y la señal de control u .

Algoritmo 33 Conjunto de filtros de Kalman para estimación de estados de un sistema lineal

Require: $A, H, B, u, x, Q, R, q, n, tru$

Ensure: xm, zm_m, x

function $[xm, zm_m, x] = \text{conjuntode filtros de Kalman}(A, H, B, u, x, Q, R, q, n, tru)$
conjunto de estados

$x_{conj} = x$

matrices de almacenamiento

$\underline{x}_V = \text{zeros}(n, q)$ //almacenamiento de estimación del conjunto de estados a-priori

$\underline{y}_V = \text{zeros}(1, q)$ //almacenamiento del observador del conjunto de estados

$zm_V = \text{zeros}(1, q)$ //almacenamiento de la salida medida del sistema en el conjunto

$Ex_V = \text{zeros}(n, q)$ //almacenamiento del error de estados del conjunto

$Ey_V = \text{zeros}(1, q)$ //almacenamiento del error de salida del conjunto

cálculos a-priori del conjunto

for $j = 1 : q$ **do**

$w(:, j) = \text{sqr}(Q)\text{randn}(n, 1)$ //ruido de la evolución del conjunto de estados

$v(:, j) = \text{sqr}(R)\text{randn}(1, 1)$ //ruido de la medición del conjunto de estados

$\underline{x} = A(x_{conj}(:, j)) + Bu + w(:, j)$ //estimación del conjunto de estados a-priori

$zm = Htru + v(:, j)$ //salida medida del sistema real en el conjunto

$\underline{y} = H\underline{x}$ //vector de observación del conjunto de estados

$\underline{x}_V(:, j) = \underline{x}$ //almacenamiento de estimación del conjunto de estados a-priori

$zm_V(:, j) = zm$ //almacenamiento de la salida medida del sistema en el conjunto

$\underline{y}_V(:, j) = \underline{y}$ //almacenamiento del observador del conjunto de estados

end for

cálculo de medias

$\underline{x}_m = \text{mean}(\underline{x}_V, 2)$ //media de la estimacion del conjunto de estados a-priori

$zm_m = \text{mean}(zm_V)$ //media de la salida medida del sistema en el conjunto

$\underline{y}_m = \text{mean}(\underline{y}_V)$ //media del vector de observacion del conjunto de estados

errores del conjunto

for $i = 1 : n$ **do**

$Ex = \underline{x}_V(i, :) - \underline{x}_m(i)$ //error de estados del conjunto

$Ex_V(i, :) = Ex$ //almacenamiento del error de estados del conjunto

end for

for $i = 1 : 1$ **do**

$Ey = \underline{y}_V(i, :) - \underline{y}_m(i)$ //error de salida del conjunto

$Ey_V(i, :) = Ey$ //almacenamiento del error de salida del conjunto

end for

Algoritmo 34 Conjunto de filtros de Kalman para estimación de estados de un sistema lineal. Continuación

covarianzas de estimación del conjunto

$$P_{xy} = Ex_V Ey_V' / (q - 1)$$

$$P_{yy} = Ey_V Ey_V' / (q - 1)$$

ganancia de Kalman

$$K = P_{xy} inv(P_{yy})$$

cálculos a-posteriori

$$x = \underline{x_V} + K(zm_V - \underline{y_V}) // \text{estimación del conjunto de estados a-posteriori}$$

$$xm = mean(x, 2) // \text{media de la estimación del conjunto de estados a-posteriori}$$

return

El algoritmo 33,34 entrega como salidas la estimación de estados xm y el conjunto de estados estimados x con el cual se cumple el criterio de recursividad, la salida medida del sistema en el conjunto zm_m a pesar de ser una salida del algoritmo no será utilizada más adelante, debido a que en este caso no se busca comparar las salidas del sistema, sino que solo se busca realizar el control de las variables de estado.

5.3. Control del sistema eólico

En la figura 5.2 es posible observar el esquema de la estimación de estados a partir del conjunto de filtros de Kalman y el control por realimentación de estados, el observador del sistema lineal, junto con su respectivo ruido son entradas del conjunto de filtros de Kalman, una vez el algoritmo realiza la estimación de estados xm , estos se multiplican por la ganancia de realimentación $-K$ para obtener el control del sistema, el cual está encargado de llevar las corrientes (variables de estado) a su punto de operación, el control actualiza la representación en espacio de estados del sistema y representa una entrada del conjunto de filtros de Kalman.

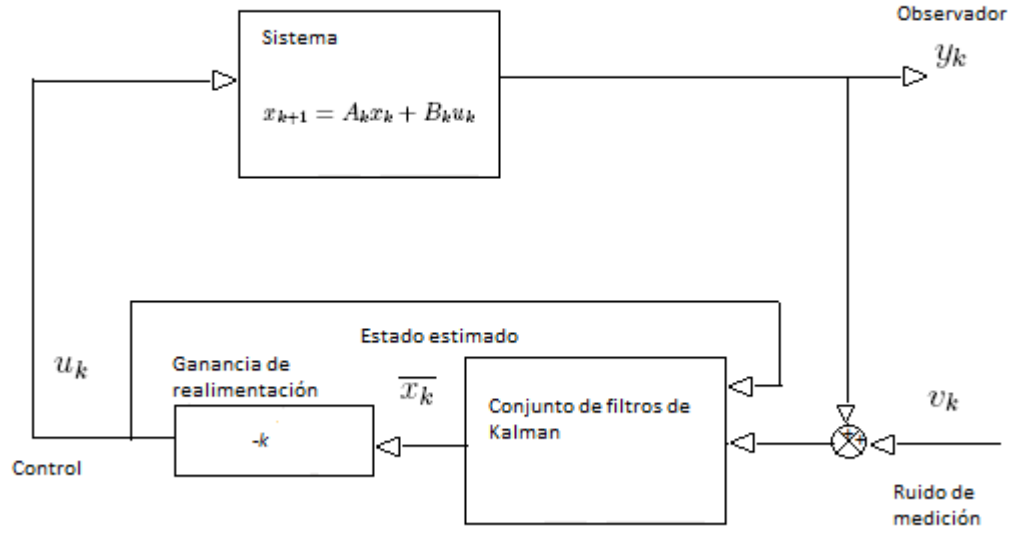


Figura 5.2: Diagrama de bloques de la estimación de estados a partir del conjunto de filtros de Kalman y del control por realimentación de estados

5.3.1. Algoritmo para el control del sistema eólico

En el algoritmo 35 se realiza el control del sistema eólico mostrado en la figura 5.1 mediante realimentación de estados a partir de los estados estimados por el conjunto de filtros de Kalman, en primer lugar se define el orden del sistema n , luego se definen los parámetros del sistema eólico, con los cuales es posible obtener la matriz de transición de estados A y la matriz de entrada B , se introducen las entradas faltantes que requiere el conjunto de filtros de Kalman (algoritmo 33,34), es decir la matriz de medición H , la señal de control u , el ruido del proceso y la medición Q y R , el número de miembros del conjunto q , los estados reales tru y la estimación del conjunto de estados inicial x .

Posteriormente se definen los polos deseados del sistema P , el número de iteraciones o mediciones N que se realizarán en el algoritmo, y dos matrices de almacenamiento, xV almacenamiento de los estados estimados y $truV$ almacenamiento de los estados reales.

Algoritmo 35 Control del sistema eólico

$n = 4$ // estados del sistema

parámetros del sistema

$w_0 = 0,4363$ // frecuencia angular base

$w = 0,2472$ // frecuencia angular arbitraria

$w_t = 0,5235$ // frecuencia angular del rotor

$b = 0,5235$ // constante

$r_s = 0,0029$ // resistencia del estator

$X_s = 0,0029$ // reactancia del estator

$r_r = 0,01$ // resistencia del rotor

$X_r = 0,01$ // reactancia del rotor

$X_m = 0,0145$ // reactancia de magnetización

$$A = \frac{w_0}{b} \begin{bmatrix} -r_s X_r & (wb + w_t X_m^2) & -r_r X_m & -w_t X_r X_m \\ -(wb + w_t X_m^2) & -r_s X_r & w_t X_r X_m & -r_r X_m \\ -r_s X_m & w_t X_s X_m & -r_r X_s & (wb - w_t X_s X_r) \\ -w_t X_s X_m & -r_s X_m & -(wb - w_t X_s X_r) & -r_r X_s \end{bmatrix}$$

// matriz de transición de estados

$$B = \frac{w_0}{b} \begin{bmatrix} -X_r & 0 & X_m & 0 \\ 0 & -X_r & 0 & X_m \\ -X_m & 0 & X_s & 0 \\ 0 & -X_m & 0 & X_s \end{bmatrix} \quad // \text{matriz de entrada}$$

$H = [0 \ 0 \ 0 \ 1]$ // matriz de medición

$Q = 0,1e - 31 \text{eye}(m)$ // ruido del proceso

$R = 0,1e - 31$ // ruido de la medición

$q = 20$ // número de miembros del conjunto

$x = \text{rand}(n, q)$ // conjunto inicial de estados estimados

$u = [1; 0.1; 0; 0]$ // señal de control

$\text{tru} = \text{ones}(n, 1)$ // estado real del sistema

$P = [0.1; 0.12; 0.15; 0.27]$ // polos de control

mediciones o iteraciones

$N = 80$

matrices de almacenamiento

$xV = \text{zeros}(n, N)$ // almacenamiento de la estimación de estados

$\text{tru}V = \text{zeros}(n, N)$ // almacenamiento del estado real del sistema

Algoritmo 36 Control del sistema eólico. Continuación

proceso iterativo

for $k = 1 : N$ **do**

$tru = Atru + Bu + \text{sqrt}(Q)\text{randn}(n, 1)$ //estado real del sistema

$truV(:, k) = tru$ //almacenamiento del estado real del sistema

$[xm, zm_m, x] = \text{conjuntode filtros de Kalman}(A, H, B, u, x, Q, R, q, n, tru)$ //realizar una iteración del conjunto de filtros de Kalman para estimación de estados de un sistema lineal

$xV(:, k) = xm$ //almacenamiento de la estimación de estados

$K = \text{place}(A, B, P)$ //ganancia de realimentación

$u = -Kxm$ //señal de control

end for

Se da paso al proceso iterativo, en el cual se calcula el estado real tru que corresponde a la representación en espacio de estados de un sistema lineal, se prosigue almacenando el estado real en su respectiva matriz.

Como paso siguiente se realiza un llamado al conjunto de filtros de Kalman para estimación de estados de un sistema lineal (algoritmo 33,34), de esta manera se obtienen los estados estimados xm , posteriormente se calcula la ganancia de realimentación K a partir de las matrices A , B y de los polos deseados P , finalmente es posible calcular la señal de control u a partir de la ganancia K y de los estados estimados xm , la señal de control u se va actualizando hasta que la realimentación de estados coloque los polos o valores propios del sistema en los lugares indicados en P . En la figura 5.2 es posible observar el control de las variables de estado, las cuales para este caso corresponden a corrientes en el estator y en el rotor de un sistema eólico.

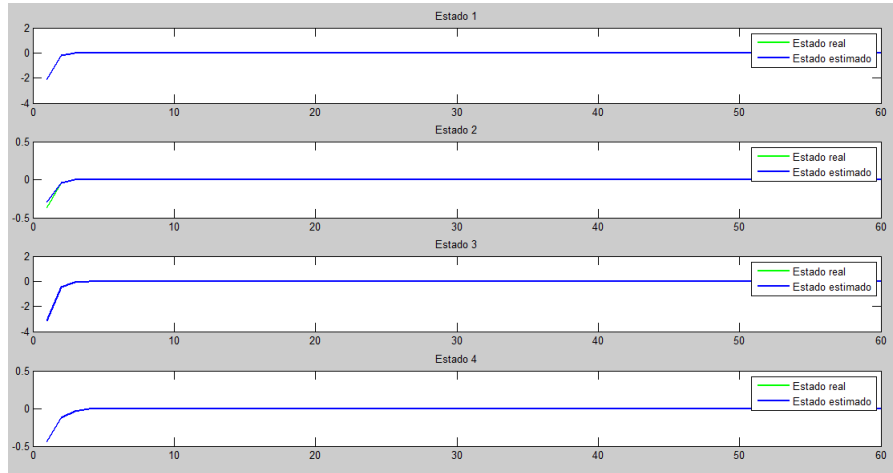


Figura 5.3: Control del sistema eólico por realimentación de estados a partir de los estados estimados por el conjunto de filtros de Kalman

6. ANÁLISIS COMPARATIVO ENTRE EL FILTRO DE KALMAN EXTENDIDO Y EL CONJUNTO DE FILTROS DE KALMAN

A continuación se realiza un análisis comparativo entre el filtro de Kalman extendido y el conjunto de filtros de Kalman respecto a estimación de estados, y estimación de parámetros.

6.1. Estimación de estados

6.1.1. Estimación de estados. Ejemplo 1

En las subsecciones 3.1.4 y 4.1.5 se realiza la estimación de estados de un carro-péndulo modelado como un sistema de cuarto orden mediante el filtro de Kalman extendido y mediante el conjunto de filtros de Kalman respectivamente, a continuación se presenta una comparación detallada respecto a la estimación de los cuatro estados del sistema.

En las figuras 6.1, 6.2, 6.3 y 6.4 es posible observar la estimación de los cuatro estados del sistema, el estado real corresponde al color verde, el estado estimado mediante el conjunto de filtros de Kalman corresponde al color azul y el estado estimado mediante el filtro de Kalman extendido corresponde al color rojo (lineal punteada).

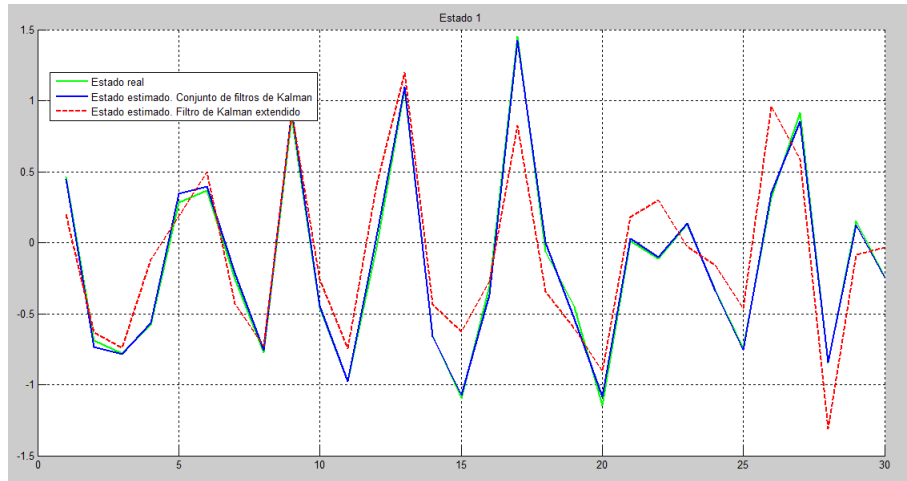


Figura 6.1: Análisis comparativo entre la estimación de estados mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman. Ejemplo 1 (sistema de cuarto orden, primer estado)

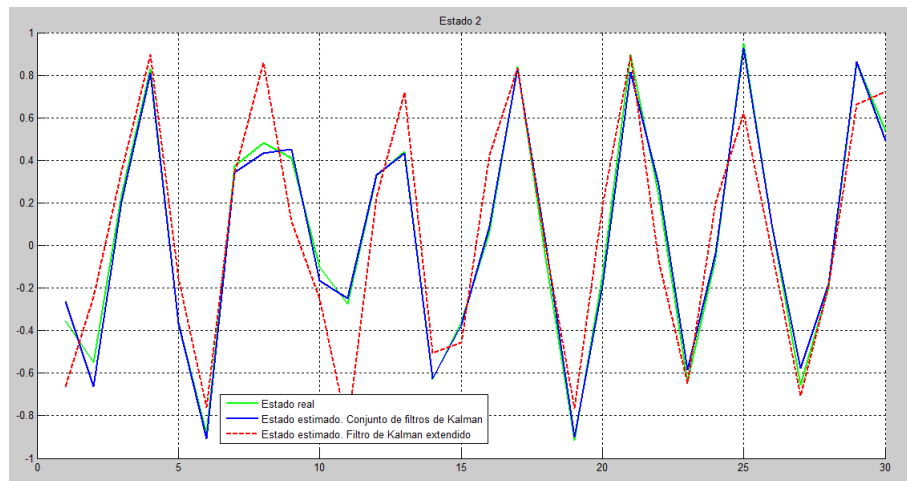


Figura 6.2: Análisis comparativo entre la estimación de estados mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman. Ejemplo 1 (sistema de cuarto orden, segundo estado)

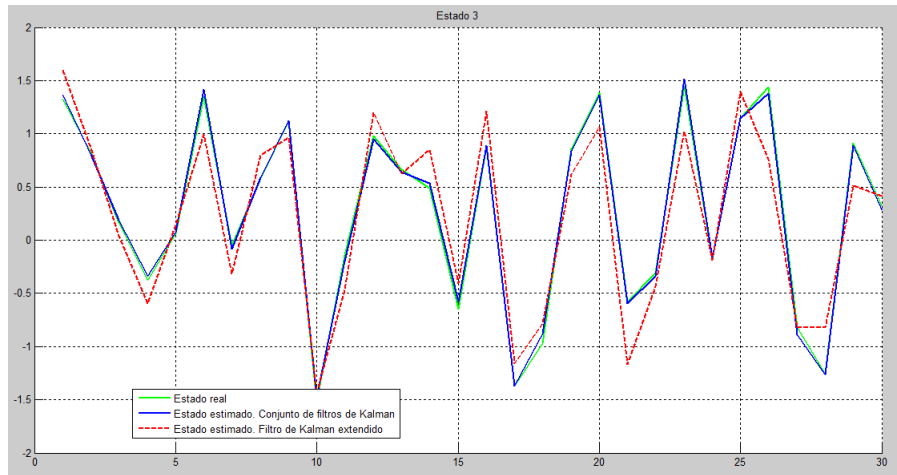


Figura 6.3: Análisis comparativo entre la estimación de estados mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman. Ejemplo 1 (sistema de cuarto orden, tercer estado)

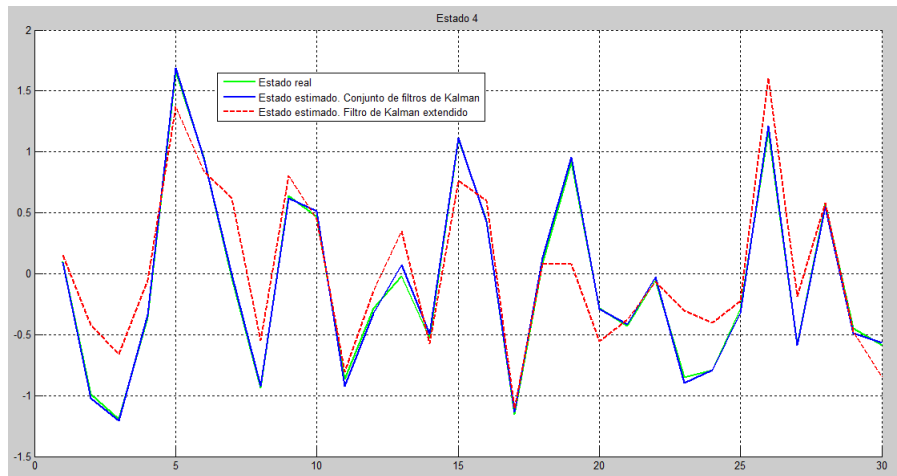


Figura 6.4: Análisis comparativo entre la estimación de estados mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman. Ejemplo 1 (sistema de cuarto orden, cuarto estado)

Claramente la estimación de estados realizada mediante el conjunto de filtros de Kalman es más optima que la realizada mediante el filtro de Kalman extendido. Para las estimaciones anteriores realizadas a partir del conjunto de filtros de Kalman fue empleado un conjunto de setenta estados.

Con el propósito de ratificar que la estimación de estados realizada a partir del conjunto de filtros de Kalman es más optima que la realizada a partir del filtro de Kalman extendido se comparan las covarianzas de error de estimación (CEE) para los dos algoritmos, se observa que el error de estimación es más grande para el caso en el cual se utiliza el filtro de Kalman extendido.

	CEE EKF	CEE EnKF
Estado 1	0.0734	0.0014
Estado 2	0.0404	0.000929
Estado 3	0.0626	0.0019
Estado 4	0.1040	0.0011

6.1.2. Estimación de estados. Ejemplo 2

En las subsecciones 3.1.5 y 4.1.6 se realizó la estimación de estados de un sistema de segundo orden mediante el filtro de Kalman extendido y mediante el conjunto de filtros de Kalman respectivamente, a continuación se realiza una comparación detallada respecto a la estimación de los dos estados del sistema.

En las figuras 6.5, y 6.6 es posible observar la estimación de los dos estados del sistema, el estado real corresponde al color verde, el estado estimado mediante el conjunto de filtros de Kalman corresponde al color azul y el estado estimado mediante el filtro de Kalman extendido corresponde al color rojo (lineal punteada).

Nuevamente se observa que la estimación de estados realizada mediante el conjunto de filtros de Kalman es más optima que la realizada mediante el filtro de Kalman extendido. Para las estimaciones anteriores realizadas a partir del conjunto de filtros de Kalman fue empleado un conjunto de cien estados.

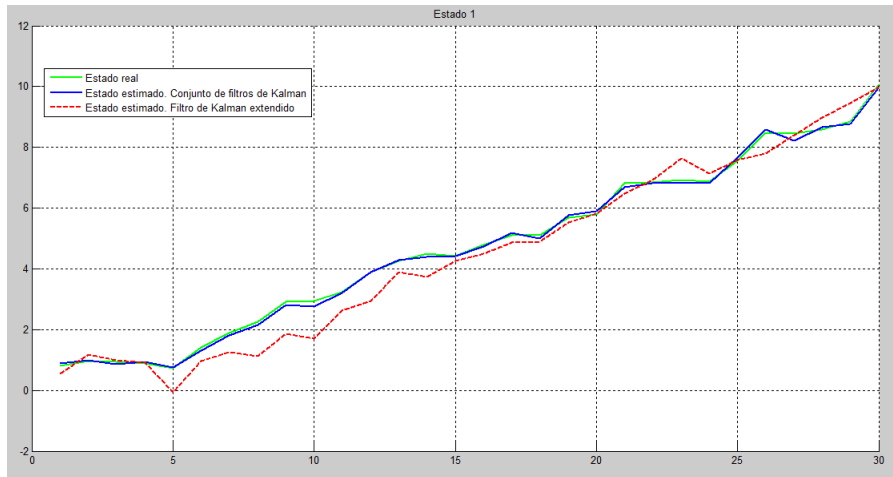


Figura 6.5: Análisis comparativo entre la estimación de estados mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman. Ejemplo 2 (sistema de segundo orden, primer estado)

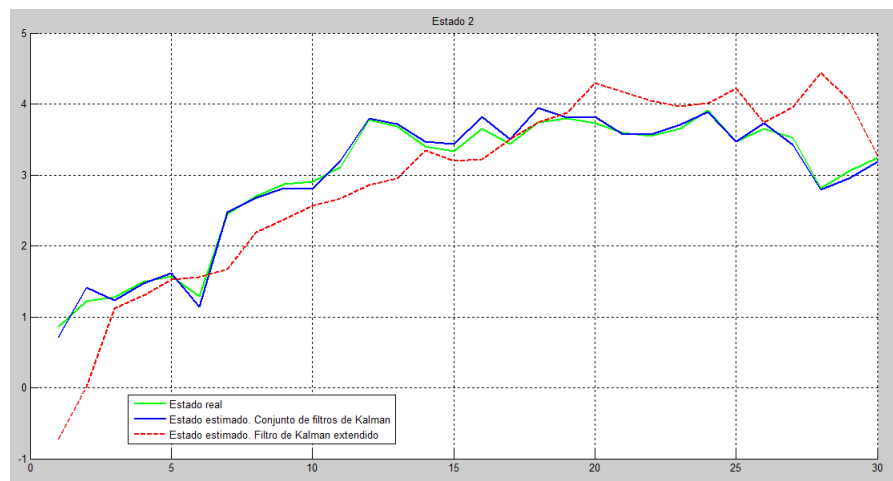


Figura 6.6: Análisis comparativo entre la estimación de estados mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman. Ejemplo 2 (sistema de segundo orden, segundo estado)

Con el propósito de ratificar que la estimación de estados realizada a partir del conjunto de filtros de Kalman es más óptima que la realizada a partir del filtro de Kalman extendido se comparan las covarianzas de error de estimación (*CEE*) para los dos algoritmos, se observa que el error de estimación es más grande para el caso en el cual se utiliza el filtro de Kalman extendido.

	CEE EKF	CEE EnKF
Estado 1	0.2622	0.0066
Estado 2	0.2236	0.0062

6.2. Estimación de parámetros

6.2.1. Estimación de parámetros. Ejemplo

En las subsecciones 3.2.4 y 4.2.2 se realizó la estimación de parámetros de una función parametrizada no lineal (función Rosenbrock) mediante el filtro de Kalman extendido y mediante el conjunto de filtros de Kalman respectivamente, a continuación se realiza una comparación respecto a la estimación de los parámetros del sistema.

En las figuras 6.7 es posible observar la estimación de los parámetros de la función Rosenbrock, los parámetros estimados mediante el conjunto de filtros de Kalman corresponden a los mostrados en color azul y verde, por otro lado los parámetros estimados mediante el filtro de Kalman extendido corresponden a los mostrados en color rojo y magenta.

Claramente es posible observar que la tasa de convergencia del conjunto de filtros de Kalman es mucho mayor a la del filtro de Kalman extendido, debido a que los parámetros estimados a partir del conjunto de filtros de Kalman alcanzan los parámetros verdaderos en la iteración treinta, en cambio los parámetros estimados a partir del filtro de Kalman extendido ni siquiera con quinientas iteraciones alcanzan los parámetros verdaderos de manera exacta ($parametroestimadouno = 0,9942, parametroestimados = 0,9883$), cabe recordar que los parámetros verdaderos son iguales a uno.

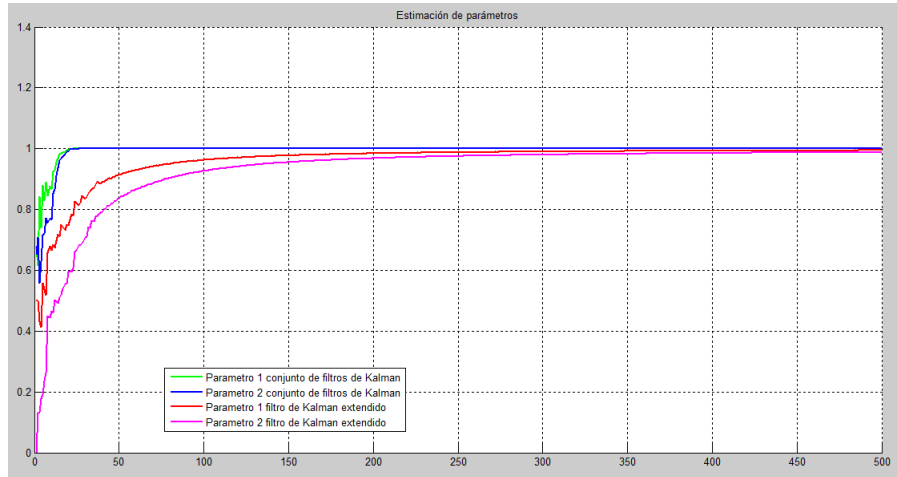


Figura 6.7: Análisis comparativo entre la estimación de parámetros mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman. Ejemplo (función Rosenbrock)

7. CONCLUSIONES

- A partir del conjunto de filtros de Kalman fue posible diseñar un algoritmo de identificación de sistemas, además fue diseñado un algoritmo para estimación de estados y estimación dual, cabe mencionar que los algoritmos mencionados pueden ser empleados tanto para sistemas lineales como no lineales, cambiando adecuadamente la representación en espacio de estados.
- Fue posible diseñar el controlador de un sistema eólico (realimentación de estados), a partir de los estados estimados por el conjunto de filtros de Kalman, al observar la figura 5.2 es posible concluir que a partir de una estimación de estados óptima es posible diseñar un controlador óptimo.
- A pesar de que los fundamentos matemáticos del filtro de Kalman, tienen cierto grado de dificultad respecto a su comprensión, la implementación del algoritmo es muy práctica, constituyendo un punto de partida para otros métodos y para abordar otras problemáticas.
- Respecto a los ejemplos presentados para estimación de estados a partir del filtro de Kalman, es posible observar que en el primer ejemplo (figura 2.2) la estimación de estados no es tan exacta como lo es en el segundo ejemplo (figura 2.5), lo anterior se debe a que en el primer ejemplo no se tiene el modelo de un sistema, sino que se tiene una serie de voltajes con ruido y la respectiva medición de estos voltajes.
- El filtro de Kalman puede ser empleado para estimar los estados o/y los parámetros de un sistema de cualquier orden, se trató de expresar lo anterior presentando más de un ejemplo por estimación, sin embargo el filtro de Kalman puede emplearse únicamente para sistemas lineales.
- Respecto a los ejemplos presentados para estimación de parámetros a partir del filtro de Kalman, es posible observar que para ambos casos (figura 2.8 y figura 2.10) la estimación se estabiliza en las primeras iteraciones, numéricamente los parámetros estimados coinciden de manera exacta con los parámetros verdaderos.

- El filtro de Kalman extendido puede ser empleado para estimar los parámetros de una función parametrizada no lineal, en el ejemplo que presenta la función Rosenbrock se realizan quinientas iteraciones (figura 3.7) y sin embargo los parámetros estimados no coinciden a la perfección con los parámetros verdaderos de la función.
- A diferencia del filtro de Kalman dual y el conjunto de filtros de Kalman dual, en los cuales se implementan dos algoritmos paralelos independientes el uno del otro, en el filtro de Kalman extendido dual se implementan dos algoritmos paralelos, en los cuales las estimaciones se complementan entre sí.
- Al realizar el análisis comparativo entre la estimación de estados mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman es posible observar que la estimación realizada mediante el conjunto de filtros de Kalman es mas optima, debido a que el filtro de Kalman extendido realiza la estimación linealizando el sistema y empleando una covarianza de estimación que corresponde a la linealización y no al sistema real.
- Al realizar el análisis comparativo entre la estimación de parámetros mediante el filtro de Kalman extendido y el conjunto de filtros de Kalman es posible observar que la tasa de convergencia del conjunto de filtros de Kalman es mucho mayor a la del filtro de Kalman extendido.
- Se realiza la estimación de estados y parámetros de una maquina síncrona modelada como un sistema de cuarto orden, comparando las figuras 3.10 y 4.9, es posible concluir que la estimación de estados realizada a partir del conjunto de filtros de Kalman es más óptima que la realizada a partir del filtro de Kalman extendido, por otro lado es posible observar que los parámetros estimados con el conjunto de filtros de Kalman coinciden con los parámetros verdaderos desde la séptima iteración (figura 4.12), en cambio los parámetros estimados con el filtro de Kalman extendido coinciden con los parámetros verdaderos desde la iteración veinte (figura 3.13).
- El conjunto de filtros de Kalman representa una gran alternativa en la estimación de estados y parámetros de sistemas no lineales, el método supera inconvenientes presentados por el filtro de Kalman extendido siendo más óptimo en varios aspectos.

7.1. Futuros trabajos de investigación

En el proyecto de grado se han desarrollado estimadores duales a partir del filtro de Kalman, el filtro de Kalman extendido y el conjunto de filtros de Kalman, a lo largo del proyecto se explicó con cuidado cada algoritmo y cada ejemplo, con el propósito de despertar el interés del lector y facilitar el entendimiento de los algoritmos. Algunos de los trabajos que quedan por hacer se nombran a continuación.

- Realizar un análisis acerca de como varia la estimación de estados y parámetros a partir del conjunto de filtros de Kalman modificando el número del conjunto.
- Implementar los algoritmos en otras problemáticas.
- Comparar el conjunto de filtros de Kalman con algoritmos como redes neuronales y algoritmo de optimización de partículas.
- Realizar el control de un sistema no lineal a partir de conjunto de filtros de Kalman.
- Realizar el análisis comparativo entre el filtro de Kalman extendido y el conjunto de filtros de Kalman respecto a carga computacional.
- Realizar el análisis comparativo entre el filtro de Kalman y el conjunto de filtros de Kalman respecto a la estimación de estados de sistemas lineales.

8. BIBLIOGRAFIA

- [1]Ljung L., “Perspectives on System Identification”. Linköpings university, Sweden, 2008.
- [2]Fu L. and Li P., “The Research Survey of System Identification Method”, Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on , 2013, 2, 397-401.
- [3]Ding F., “System identification, Part A: Introduction to the identification”, Journal of Nanjing University of Information Science and Technology (Natural Science Edition), 2011, 3, 1- 22.
- [4]Ljung L., “System Identification: Theory for the User”. Englewood Cliffs, NJ:Prentice Hall, 1987.
- [5]Garrido S., “Identificación, Estimación y Control de Sistemas No-lineales mediante RGO”. Doctorado [tesis], Universidad Carlos III de Madrid, 1999.
- [6] Xiuqin Deng., “System Identification Based on Particle Swarm Optimization Algorithm”. Computational Intelligence and Security, 2009. CIS '09. International Conference on , 2009, 1, 259-263.
- [7] Bobál V.; Böhm J.; Fessl J.; and Macháček J., “Digital Self-tuning Controllers, Algorithms, Implementation and Applications”. Springer Science and Business Media, 2006.
- [8] Evensen G., “The ensemble Kalman filter for combined state and parameter estimation”, Control Systems, IEEE , 2009, 29, 83-104.
- [9]Ljung L., “System Identification Toolbox: for use with MATLAB; computation, visualization, programming; user’s guide; version 6”. MathWorks Incorporated, 2004.

- [10] Lam Q. M., “Extended Kalman Filter and Neural Network Filter: Who Outperforms Whom in System identification?”. Aerospace Control Systems, 1993. Proceedings. The First IEEE Regional Conference on , 1993, 810-814.
- [11] Hsieh B. and Kot C. and Srinivasan M., “System Identification Methods and Applications”. Argonne National Lab., USA, 1984.
- [12] Xiu-ying L. and Zhi-gang H., “Advances in nonlinear system identification”. Techniques of Automation and Application, 2004, 23, 5-7.
- [13] Eberhart R.C. and Shi Y., “Particle swarm optimization: developments, applications and resources”. Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 2001, 1, 81-86.
- [14] Gevers M., “A personal view on the development of system identification: A 30-year journey through an exciting field”. Control Systems, IEEE, 2006, 26, 93-105.
- [15] Åström, Karl J. and Bohlin T., “Numerical identification of linear dynamic systems from normal operating records”. Theory of self-adaptive control systems, Springer, 1966, 96-111.
- [16] Ho B. and Kalman R.E., “Effective construction of linear state-variable models from input/output functions”. at-Automatisierungstechnik, 1966, 14, 545-548.
- [17] Akaike H., “Stochastic theory of minimal realization”. Automatic Control, IEEE Transactions on IEEE, 1974, 19, 667-674.
- [18] Bertelsen A., Hansen T., Pedersen R., Schwensen J. and Senthuran S., “Modelling and Control of a Small Scale Wind Turbine”. Aalborg University, Denmark, 2011.
- [19] Haykin S.S., “Kalman filtering and neural networks”. Wiley Online Library, 2001.
- [20] Al-Ghossini H., Locment F. and Sechilariu M., “Experimental comparison of small wind turbine vector control with and without position sensor-Extended Kalman filter application”. Power Electronics and Applications (EPE), 2013 15th European Conferen-

ce on , 2013, 1-9.

[21]Gillijns S., Mendoza O.B., Chandrasekar J., De Moor B., Bernstein D., and Ridley A., “What is the ensemble Kalman filter and how well does it work?”. American Control Conference, 2006, 6-pp.

[22]Kalman R.E., “A new approach to linear filtering and prediction problems”. Journal of Fluids Engineering, American Society of Mechanical Engineers, 1960, 82, 35-45.

[23]Welch G., Bishop G., “An Introduction to the Kalman Filter”. 2006, University of North Carolina: Chapel Hill, North Carolina, US.

[24]Ridout M.S., “Statistical Applications of the Complex-Step Method of Numerical Differentiation”, The American Statistician, 2009, 63.

[25]Ogata K., “Ingeniería de control moderna”. Pearson Educación, 2003.

[26] Grewal M., Andrews A., “ Kalman Filtering Teory and Practice Using MATLAB”, John Wiley and Sons, 2001.

[27]Nelson L., Stear E., “The simultaneous on-line estimation of parameters and states in linear systems”. Automatic Control, IEEE Transactions on, IEEE, 1976, 21, 94-98.

[28]Ruvalcaba F., Llama M., Santibáñez V., “Control de un Carro Péndulo como un Sistema de Cuarto Orden”, Congreso Anual 2010 de la Asociación de México de Control Automático. Puerto Vallarta, Jalisco, México.

[29]Hai L., Hailan Y., Yehua L., Maode Y., “Design of dual extended Kalman filter for permanent magnet synchronous motor”. Control Conference (CCC), 2013 32nd Chinese, 2013, 1729-1734.

[30]Srikanthan R., Amirthanathan G., Kuczera G., “Real-time flood forecasting using ensemble Kalman filter”. MODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, 2007, 1789-1795.

- [31] Bavdekar V., Prakash J., Shah S., Gopaluni R., “Constrained dual ensemble Kalman filter for state and parameter estimation”. American Control Conference (ACC), 2013, 3093-3098.
- [32] Muhandó E., Senjyu T., Funabashi T., “Model fidelity prerequisites for variable speed pitch-regulated wecs with state-feedback control”. Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on, 2008, 2353-2358.